User's Manual

32

# RX111 Group

## User's Manual: Hardware

RENESAS 32-Bit MCU
RX Family / RX100 Series



Renesas Electronics
www.renesas.com

Rev.1.00 Jun 2013

# Notice

1. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation of these circuits, software, and information in the design of your equipment. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from the use of these circuits, software, or information.
2. Renesas Electronics has used reasonable care in preparing the information included in this document, but Renesas Electronics does not warrant that such information is error free. Renesas Electronics assumes no liability whatsoever for any damages incurred by you resulting from errors in or omissions from the information included herein.
3. Renesas Electronics does not assume any liability for infringement of patents, copyrights, or other intellectual property rights of third parties by or arising from the use of Renesas Electronics products or technical information described in this document. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
4. You should not alter, modify, copy, or otherwise misappropriate any Renesas Electronics product, whether in whole or in part. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from such alteration, modification, copy or otherwise misappropriation of Renesas Electronics product.
5. Renesas Electronics products are classified according to the following two quality grades: “Standard” and “High Quality”. The recommended applications for each Renesas Electronics product depends on the product’s quality grade, as indicated below.

   “Standard”: Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; and industrial robots etc.

   “High Quality”: Transportation equipment (automobiles, trains, ships, etc.); traffic control systems; anti-disaster systems; anti-crime systems; and safety equipment etc.

   Renesas Electronics products are neither intended nor authorized for use in products or systems that may pose a direct threat to human life or bodily injury (artificial life support devices or systems, surgical implantations etc.), or may cause serious property damages (nuclear reactor control systems, military equipment etc.). You must check the quality grade of each Renesas Electronics product before using it in a particular application. You may not use any Renesas Electronics product for any application for which it is not intended. Renesas Electronics shall not be in any way liable for any damages or losses incurred by you or third parties arising from the use of any Renesas Electronics product for which the product is not intended by Renesas Electronics.
6. You should use the Renesas Electronics products described in this document within the range specified by Renesas Electronics, especially with respect to the maximum rating, operating supply voltage range, movement power voltage range, heat radiation characteristics, installation and other product characteristics. Renesas Electronics shall have no liability for malfunctions or damages arising out of the use of Renesas Electronics products beyond such specified ranges.
7. Although Renesas Electronics endeavors to improve the quality and reliability of its products, semiconductor products have specific characteristics such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Further, Renesas Electronics products are not subject to radiation resistance design. Please be sure to implement safety measures to guard them against the possibility of physical injury, and injury or damage caused by fire in the event of the failure of a Renesas Electronics product, such as safety design for hardware and software including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult, please evaluate the safety of the final products or systems manufactured by you.
8. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. Please use Renesas Electronics products in compliance with all applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive. Renesas Electronics assumes no liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
9. Renesas Electronics products and technology may not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations. You should not use Renesas Electronics products or technology described in this document for any purpose relating to military applications or use by the military, including but not limited to the development of weapons of mass destruction. When exporting the Renesas Electronics products or technology described in this document, you should comply with the applicable export control laws and regulations and follow the procedures required by such laws and regulations.
10. It is the responsibility of the buyer or distributor of Renesas Electronics products, who distributes, disposes of, or otherwise places the product with a third party, to notify such third party in advance of the contents and conditions set forth in this document, Renesas Electronics assumes no responsibility for any losses incurred by you or third parties as a result of unauthorized use of Renesas Electronics products.
11. This document may not be reproduced or duplicated in any form, in whole or in part, without prior written consent of Renesas Electronics.
12. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products, or if you have any other inquiries.

(Note 1) “Renesas Electronics” as used in this document means Renesas Electronics Corporation and also includes its majority-owned subsidiaries.

(Note 2) “Renesas Electronics product(s)” means any product developed or manufactured by or for Renesas Electronics.

(2012.4)

## NOTES FOR CMOS DEVICES

(1) VOLTAGE APPLICATION WAVEFORM AT INPUT PIN: Waveform distortion due to input noise or a reflected wave may cause malfunction. If the input of the CMOS device stays in the area between VIL (MAX) and VIH (MIN) due to noise, etc., the device may malfunction. Take care to prevent chattering noise from entering the device when the input level is fixed, and also in the transition period when the input level passes through the area between VIL (MAX) and VIH (MIN).

(2) HANDLING OF UNUSED INPUT PINS: Unconnected CMOS device inputs can be cause of malfunction. If an input pin is unconnected, it is possible that an internal input level may be generated due to noise, etc., causing malfunction. CMOS devices behave differently than Bipolar or NMOS devices. Input levels of CMOS devices must be fixed high or low by using pull-up or pull-down circuitry. Each unused pin should be connected to VDD or GND via a resistor if there is a possibility that it will be an output pin. All handling related to unused pins must be judged separately for each device and according to related specifications governing the device.

(3) PRECAUTION AGAINST ESD: A strong electric field, when exposed to a MOS device, can cause destruction of the gate oxide and ultimately degrade the device operation. Steps must be taken to stop generation of static electricity as much as possible, and quickly dissipate it when it has occurred. Environmental control must be adequate. When it is dry, a humidifier should be used. It is recommended to avoid using insulators that easily build up static electricity. Semiconductor devices must be stored and transported in an anti-static container, static shielding bag or conductive material. All test and measurement tools including work benches and floors should be grounded. The operator should be grounded using a wrist strap. Semiconductor devices must not be touched with bare hands. Similar precautions need to be taken for PW boards with mounted semiconductor devices.

(4) STATUS BEFORE INITIALIZATION: Power-on does not necessarily define the initial status of a MOS device. Immediately after the power source is turned ON, devices with reset functions have not yet been initialized. Hence, power-on does not guarantee output pin levels, I/O settings or contents of registers. A device is not initialized until the reset signal is received. A reset operation must be executed immediately after power-on for devices with reset functions.

(5) POWER ON/OFF SEQUENCE: In the case of a device that uses different power supplies for the internal operation and external interface, as a rule, switch on the external power supply after switching on the internal power supply. When switching the power supply off, as a rule, switch off the external power supply and then the internal power supply. Use of the reverse power on/off sequences may result in the application of an overvoltage to the internal elements of the device, causing malfunction and degradation of internal elements due to the passage of an abnormal current. The correct power on/off sequence must be judged separately for each device and according to related specifications governing the device.

(6) INPUT OF SIGNAL DURING POWER OFF STATE : Do not input signals or an I/O pull-up power supply while the device is not powered. The current injection that results from input of such a signal or I/O pull-up power supply may cause malfunction and the abnormal current that passes in the device at this time may cause degradation of internal elements. Input of signals during the power off state must be judged separately for each device and according to related specifications governing the device.

## General Precautions in the Handling of MPU/MCU Products

The following usage notes are applicable to all MPU/MCU products from Renesas. For detailed usage notes on the products covered by this document, refer to the relevant sections of the document as well as any technical updates that have been issued for the products.

1. Handling of Unused Pins

   Handle unused pins in accord with the directions given under Handling of Unused Pins in the manual.

   — The input pins of CMOS products are generally in the high-impedance state. In operation with an unused pin in the open-circuit state, extra electromagnetic noise is induced in the vicinity of LSI, an associated shoot-through current flows internally, and malfunctions occur due to the false recognition of the pin state as an input signal become possible. Unused pins should be handled as described under Handling of Unused Pins in the manual.

2. Processing at Power-on

   The state of the product is undefined at the moment when power is supplied.

   — The states of internal circuits in the LSI are indeterminate and the states of register settings and pins are undefined at the moment when power is supplied.
   In a finished product where the reset signal is applied to the external reset pin, the states of pins are not guaranteed from the moment when power is supplied until the reset process is completed.
   In a similar way, the states of pins in a product that is reset by an on-chip power-on reset function are not guaranteed from the moment when power is supplied until the power reaches the level at which resetting has been specified.

3. Prohibition of Access to Reserved Addresses

   Access to reserved addresses is prohibited.

   — The reserved addresses are provided for the possible future expansion of functions. Do not access these addresses; the correct operation of LSI is not guaranteed if they are accessed.

4. Clock Signals

   After applying a reset, only release the reset line after the operating clock signal has become stable. When switching the clock signal during program execution, wait until the target clock signal has stabilized.

   — When the clock signal is generated with an external resonator (or from an external oscillator) during a reset, ensure that the reset line is only released after full stabilization of the clock signal. Moreover, when switching to a clock signal produced with an external resonator (or by an external oscillator) while program execution is in progress, wait until the target clock signal is stable.

5. Differences between Products

   Before changing from one product to another, i.e. to one with a different part number, confirm that the change will not lead to problems.

   — The characteristics of MPU/MCU in the same group but having different part numbers may differ because of the differences in internal memory capacity and layout pattern. When changing to products of different part numbers, implement a system-evaluation test for each of the products.

# Specification Differences between Products

There are the following specification differences in these MCU products depending on the package.

**Table 1 Specification Differences Depending on Packages**

| Chapter | | Specification Differences | |
|---|---|---|---|
| | | Products with 40 pins or less | Products with 48 pins or more |
| 9.Clock Generation Circuit | 9.8.4 Notes on Sub-Clock | Although sub-clock oscillator pins are not available, the sub-clock circuit must be initialized at a cold start. | At a cold start, initialize the sub-clock control circuit regardless of whether the sub-clock is in use or not. |

# How to Use This Manual

## 1. Objective and Target Users

This manual was written to explain the hardware functions and electrical characteristics of this LSI to the target users, i.e. those who will be using this LSI in the design of application systems. Target users are expected to understand the fundamentals of electrical circuits, logic circuits, and microcomputers.
This manual is organized in the following items: an overview of the product, descriptions of the CPU, system control functions, and peripheral functions, electrical characteristics of the device, and usage notes.

When designing an application system that includes this LSI, take all points to note into account.
Points to note are given in their contexts and at the final part of each section, and in the section giving usage notes.

The list of revisions is a summary of major points of revision or addition for earlier versions. It does not cover all revised items. For details on the revised points, see the actual locations in the manual.

The following documents have been prepared for the RX111 Group. Before using any of the documents, please visit our website to verify that you have the most up-to-date available version of the document.

| Document Type | Contents | Document Title | Document No. |
|---|---|---|---|
| DataSheet | Overview of hardware and electrical characteristics | — | — |
| User's Manual: Hardware | Hardware specifications (pin assignments, memory maps, peripheral specifications, electrical characteristics, and timing charts) and descriptions of operation | RX111 Group User's Manual: Hardware | This document |
| User's Manual: Software | Detailed descriptions of the CPU and instruction set | RX Family User's Manual: Software | REJ09B0435 |
| Application Note | Examples of applications and sample programs | — | — |
| Renesas Technical Update | Preliminary report on the specifications of a product, document, etc. | — | — |

## 2. Description of Registers

Each register description includes a bit chart, illustrating the arrangement of bits, and a table of bits, describing the meanings of the bit settings. The standard format and notation for bit charts and tables are described below.

### X.X.X ... Register

Address(es): xxxx xxxxh

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | ... | ... | ... | — | — | — | ... |
| Value after reset: | x | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b0 | ... 0 | ... Bit | 0: .....<br>1: Setting prohibited | R/W |
| b3 to b1 | — | Reserved | The read value is 0. The write value should be 0. | R/W |
| b4 | ... 4 | ... Bit | 0: .....<br>1: ..... | R |
| b6, b5 | ... [1:0] | ... Bit | 0 0: .....<br>0 1: .....<br>Settings other than above are prohibited. | R/(W)* |
| b7 | — | Reserved | The read value is undefined. Writing to this bit has no effect. | R |

(1) R/W: The bit or field is readable and writable.
R/(W): The bit or field is readable and writable. However, writing to this bit or field has some limitations. For details on the limitations, see the description or notes of respective registers.
R: The bit or field is readable. Writing to this bit or field has no effect.

(2) Reserved. Make sure to use the specified value when writing to this bit or field; otherwise, the correct operation is not guaranteed.

(3) Setting prohibited. The correct operation is not guaranteed if such a setting is performed.

# 3. List of Abbreviations and Acronyms

| Abbreviation | Full Form |
|---|---|
| ACIA | Asynchronous Communication Interface Adapter |
| bps | bits per second |
| CRC | Cyclic Redundancy Check |
| DMA | Direct Memory Access |
| DMAC | Direct Memory Access Controller |
| GSM | Global System for Mobile Communications |
| Hi-Z | High Impedance |
| IEBus | Inter Equipment Bus |
| I/O | Input/Output |
| IrDA | Infrared Data Association |
| LSB | Least Significant Bit |
| MSB | Most Significant Bit |
| NC | Non-Connect |
| PLL | Phase Locked Loop |
| PWM | Pulse Width Modulation |
| SIM | Subscriber Identity Module |
| UART | Universal Asynchronous Receiver/Transmitter |
| VCO | Voltage Controlled Oscillator |

# Contents

R01UH0365EJ0100
Rev.1.00
Jun 14, 2013

# 32 MHz 32-bit RX MCUs, 50 DMIPS, up to 128 Kbytes of flash memory, USB 2.0 full-speed host/function/OTG, up to 6 comms channels, 12-bit A/D, 8-bit D/A, RTC

## Features

### ■ 32-bit RX CPU core
- 32 MHz maximum operating frequency
  Capable of 50 DMIPS when operating at 32 MHz
- Accumulator handles 64-bit results (for a single instruction) from 32-bit × 32-bit operations
- Multiplication and division unit handles 32-bit × 32-bit operations (multiplication instructions take one CPU clock cycle)
- Fast interrupt
- CISC Harvard architecture with five-stage pipeline
- Variable-length instruction format, ultra-compact code
- On-chip debugging circuit

### ■ Low power consumption functions
- Operation from a single 1.8 to 3.6 V supply
- Three low power consumption modes

### ■ On-chip flash memory for code, no wait states
- Operation at 32 MHz, read cycle of 31.25 ns
- No wait states for reading at full CPU speed
- 16 to 128 Kbyte capacities
- Programmable at 1.8 V
- For instructions and operands

### ■ On-chip data flash memory
- 8 Kbytes
  1,000,000 Erase/Write cycles (typ.)
- BGO (Background Operation)

### ■ On-chip SRAM, no wait states
- 8 to 16 Kbyte capacities

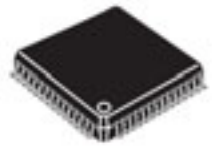
### ■ Data transfer controller (DTC)
- Four transfer modes
- Transfer can be set for each interrupt source.

### ■ Event link controller (ELC)
- Module operation can be initiated by event signals without going through interrupts.
- Link operation between modules is possible while the CPU is sleeping.

### ■ Reset and power supply voltage management
- Six types including Power-On Reset (POR)
- Low voltage detection (LVD) with voltage settings

### ■ Clock functions
- External clock input frequency: Up to 20 MHz
- Main clock oscillator frequency: 1 to 20 MHz
- Sub-clock oscillator frequency: 32.768 kHz
- PLL circuit input: 4 to 8 MHz
- Low-speed on-chip oscillator: 4 MHz
- High-speed on-chip oscillator: 32 MHz
- IWDT-dedicated on-chip oscillator: 15 kHz
- Generate a dedicated 32.768-kHz clock for the RTC
- On-chip clock frequency accuracy measurement circuit (CAC)

### ■ Realtime clock (RTC)
- 30-second, leap year, and error adjustment functions
- Calendar count mode or binary count mode selectable
- Capable of initiating exit from software standby mode



PLQP0064KB-A 10 × 10 mm, 0.5 mm pitch
PLQP0064GA-A 14 × 14 mm, 0.8 mm pitch
PLQP0048KB-A 7 × 7 mm, 0.5 mm pitch

PWQN0048KB-A 7 × 7 mm, 0.50 mm pitch
PWQN0040KC-A 6 × 6 mm, 0.50 mm pitch

PWLG0064KA-A 5 × 5 mm, 0.5 mm pitch
PWLG0036KA-A 4 × 4 mm, 0.5 mm pitch

### ■ Independent watchdog timer (IWDT)
- 15-kHz on-chip oscillator produces a dedicated clock signal to drive IWDT operation.

### ■ On-chip functions for IEC 60730 compliance
- Clock frequency accuracy measurement circuit, IWDT, functions to assist in RAM testing, etc.

### ■ Up to six channels for communication
- USB: USB 2.0 host (32 Kbyte or more ROM)/function/ On-The-Go (OTG) (one channel), full-speed = 12 Mbps, low-speed = 1.5 Mbps, isochronous transfer, and BC (Battery Charger) supported
- SCI: Asynchronous mode, clock synchronous mode, smart card interface (up to three channels)
- $I^2C$ bus interface: Transfer at up to 400 kbps, capable of SMBus operation (one channel)
- RSPI (one channel)

### ■ Up to 8 extended-function timers
- 16-bit MTU: Input capture/output compare, complementary PWM output, phase counting mode (six channels)
- 16-bit CMT (two channels)

### ■ 12-bit A/D converter
- Up to 14 channels
- 1.0 µs minimum conversion speed
- Double trigger (data duplication) function for motor control

### ■ 8-bit D/A converter
- Two channels (for 64 pins only)

### ■ Temperature sensor

### ■ General I/O ports
- 5-V tolerant, open drain, input pull-up

### ■ Multi-function pin controller (MPC)
- Multiple I/O pins can be selected for peripheral functions.

### ■ Operating temperature range
- –40 to +85°C
- –40 to +105°C

# 1. Overview

## 1.1 Outline of Specifications

Table 1.1 lists the specifications, and Table 1.2 gives a comparison of the functions of the products in different packages.
Table 1.1 is for products with the greatest number of functions, so the number of peripheral modules and channels will differ in accordance with the package type. For details, see Table 1.2, Comparison of Functions for Different Packages.

**Table 1.1 Outline of Specifications (1/3)**

| Classification | Module/Function | Description |
|---|---|---|
| CPU | CPU | • Maximum operating frequency: 32 MHz<br>• 32-bit RX CPU<br>• Minimum instruction execution time: One instruction per clock cycle<br>• Address space: 4-Gbyte linear<br>• Register set<br>General purpose: Sixteen 32-bit registers<br>Control: Eight 32-bit registers<br>Accumulator: One 64-bit register<br>• Basic instructions: 73<br>• DSP instructions: 9<br>• Addressing modes: 10<br>• Data arrangement<br>Instructions: Little endian<br>Data: Selectable as little endian or big endian<br>• On-chip 32-bit multiplier: 32-bit × 32-bit → 64-bit<br>• On-chip divider: 32-bit ÷ 32-bit → 32 bits<br>• Barrel shifter: 32 bits |
| Memory | ROM | • Capacity: 16 K /32 K /64 K /96 K /128 Kbytes<br>• 32 MHz, no-wait memory access<br>• Programming/erasing method:<br>Serial programming (asynchronous serial communication/USB communication), self-programming |
| | RAM | • Capacity: 8 K /10 K /16 Kbytes<br>• 32 MHz, no-wait memory access |
| | E2 DataFlash | • Capacity: 8 Kbytes<br>• Number of erase/write cycles: 1,000,000 (typ) |
| MCU operating mode | | Single-chip mode |
| Clock | Clock generation circuit | • Main clock oscillator, sub-clock oscillator, low-speed on-chip oscillator, high-speed on-chip oscillator, PLL frequency synthesizer, and IWDT-dedicated on-chip oscillator<br>• Oscillation stop detection: Available<br>• Clock frequency accuracy measurement circuit (CAC)<br>• Independent settings for the system clock (ICLK), peripheral module clock (PCLK), and FlashIF clock (FCLK)<br>The CPU and system sections such as other bus masters run in synchronization with the system clock (ICLK): 32 MHz (at max.)<br>Peripheral modules run in synchronization with the PCLK: 32 MHz (at max.)<br>The flash peripheral circuit runs in synchronization with the FCLK: 32 MHz (at max.)<br>• The ICLK frequency can only be set to FCLK, PCLKB, or PCLKD multiplied by n (n: 1, 2, 4, 8, 16, 32, 64). |
| Resets | | RES# pin reset, power-on reset, voltage monitoring reset, independent watchdog timer reset, and software reset |
| Voltage detection | Voltage detection circuit (LVDAa) | • When the voltage on VCC falls below the voltage detection level, an internal reset or internal interrupt is generated.<br>Voltage detection circuit 1 is capable of selecting the detection voltage from 10 levels<br>Voltage detection circuit 2 is capable of selecting the detection voltage from 4 levels |
| Low power consumption | Low power consumption functions | • Module stop function<br>• Three low power consumption modes<br>Sleep mode, deep sleep mode, and software standby mode |
| | Function for lower operating power consumption | • Operating power control modes<br>High-speed operating mode, middle-speed operating mode, and low-speed operating mode |
| Interrupt | Interrupt controller (ICUb) | • Interrupt vectors: 82<br>• External interrupts: 9 (NMI, IRQ0 to IRQ7 pins)<br>• Non-maskable interrupts: 4 (NMI pin, voltage monitoring 1 interrupt, voltage monitoring 2 interrupt, and IWDT interrupt)<br>• 16 levels specifiable for the order of priority |

**Table 1.1 Outline of Specifications (2/3)**

| Classification | Module/Function | Description |
|---|---|---|
| DMA | Data transfer controller (DTCa) | • Transfer modes: Normal transfer, repeat transfer, and block transfer<br>• Activation sources: Interrupts<br>• Chain transfer function |
| I/O ports | General I/O ports | 64-pin /48-pin /40-pin /36-pin<br>• I/O: 46/30/24/20<br>• Input: 2/2/1/1<br>• Pull-up resistors: 38/24/19/16<br>• Open-drain outputs: 34/24/19/16<br>• 5-V tolerance: 4/4/4/4 |
| Event link controller (ELC) | | • Event signals of 35 types can be directly connected to the module<br>• Operations of timer modules are selectable at event input<br>• Capable of event link operation for port B |
| Multi-function pin controller (MPC) | | Capable of selecting the input/output function from multiple pins |
| Timers | Multi-function timer pulse unit 2 (MTU2a) | • (16 bits × 6 channels) × 1 unit<br>• Time bases for the six 16-bit timer channels can be provided via up to 16 pulse-input/output lines and three pulse-input lines<br>• Select from among eight or seven counter-input clock signals for each channel (PCLK/1, PCLK/4, PCLK/16, PCLK/64, PCLK/256, PCLK/1024, MTCLKA, MTCLKB, MTCLKC, MTCLKD) other than channel 5, for which only four signals are available.<br>• Input capture function<br>• 21 output compare/input capture registers<br>• Pulse output mode<br>• Complementary PWM output mode<br>• Reset synchronous PWM mode<br>• Phase-counting mode<br>• Capable of generating conversion start triggers for the A/D converter |
| | Port output enable 2 (POE2a) | Controls the high-impedance state of the MTU's waveform output pins |
| | Compare match timer (CMT) | • (16 bits × 2 channels) × 1 unit<br>• Select from among four clock signals (PCLK/8, PCLK/32, PCLK/128, PCLK/512) |
| | Independent watchdog timer (IWDTa) | • 14 bits × 1 channel<br>• Count clock: Dedicated low-speed on-chip oscillator for the IWDT<br>Frequency divided by 1, 16, 32, 64, 128, or 256 |
| | Realtime clock (RTCA) | • Clock source: Sub-clock<br>• Calendar count mode or binary count mode selectable<br>• Interrupts: Alarm interrupt, periodic interrupt, and carry interrupt |
| Communication functions | Serial communications interfaces (SCIe, SCIf) | • 3 channels (channel 1, 5: SCIe, channel 12: SCIf)<br>• Serial communications modes: Asynchronous, clock synchronous, and smart card interface<br>• On-chip baud rate generator allows selection of the desired bit rate<br>• Choice of LSB-first or MSB-first transfer<br>• Average transfer rate clock can be input from MTU2 timers<br>• Simple $I^2C$<br>• Simple SPI<br>• Master/slave mode supported (SCIf only)<br>• Start frame and information frame are included (SCIf only)<br>• Start-bit detection in asynchronous mode: Low level or falling edge is selectable |
| | $I^2C$ bus interface (RIIC) | • 1 channel<br>• Communications formats:<br>$I^2C$ bus format/SMBus format<br>• Master mode or slave mode selectable<br>• Supports fast mode |
| | Serial peripheral interface (RSPI) | • 1 channel<br>• Transfer facility<br>Using the MOSI (master out, slave in), MISO (master in, slave out), SSL (slave select), and RSPI clock (RSPCK) signals enables serial transfer through SPI operation (four lines) or clock-synchronous operation (three lines)<br>• Capable of handling serial transfer as a master or slave<br>• Data formats<br>• Choice of LSB-first or MSB-first transfer<br>The number of bits in each transfer can be changed to 8, 9, 10, 11, 12, 13, 14, 15, 16, 20, 24, or 32 bits.<br>128-bit buffers for transmission and reception<br>Up to four frames can be transmitted or received in a single transfer operation (with each frame having up to 32 bits)<br>• Double buffers for both transmission and reception |

**Table 1.1 Outline of Specifications (3/3)**

| Classification | Module/Function | Description |
|---|---|---|
| Communication function | USB 2.0 host/function module (USBc) | • USB Device Controller (UDC) and transceiver for USB 2.0 are incorporated.<br>• Host (32-Kbyte or more ROM)/function module: 1 port<br>• Compliant with USB version 2.0<br>• Transfer speed: Full-speed (12 Mbps), low-speed (1.5 Mbps)<br>• OTG (ON-The-Go) is supported.<br>• Isochronous transfer is supported.<br>• BC (Battery Charger) is supported. |
| 12-bit A/D converter (S12ADb) | | • 1 unit (1 unit × 14 channels)<br>• 12-bit resolution<br>• Minimum conversion time: 1.0 µs per channel when the ADCLK is operating at 32 MHz<br>• Operating modes<br>Scan mode (single scan mode, continuous scan mode, and group scan mode)<br>• Double trigger mode (duplication of A/D conversion data)<br>• A/D conversion start conditions<br>A software trigger, a trigger from a timer (MTU), an external trigger signal, or ELC |
| Temperature sensor (TEMPSa) | | • 1 channel<br>• The voltage of the temperature is converted into a digital value by the 12-bit A/D converter. |
| D/A converter (DA) | | • 2 channels<br>• 8-bit resolution<br>• Output voltage: 0 V to VCC |
| CRC calculator (CRC) | | • CRC code generation for arbitrary amounts of data in 8-bit units<br>• Select any of three generating polynomials:<br>$X^8 + X^2 + X + 1$, $X^{16} + X^{15} + X^2 + 1$, or $X^{16} + X^{12} + X^5 + 1$<br>• Generation of CRC codes for use with LSB-first or MSB-first communications is selectable. |
| Data operation circuit (DOC) | | Comparison, addition, and subtraction of 16-bit data |
| Power supply voltages/Operating frequencies | | VCC = 1.8 to 2.4 V: 8 MHz, VCC = 2.4 to 2.7 V: 16 MHz, VCC = 2.7 to 3.6 V: 32 MHz |
| Supply current | | 3.2 mA at 32 MHz (typ.) |
| Operating temperature range | | D version: –40 to +85°C, G version: –40 to +105°C |
| Packages | | 64-pin LFQFP (PLQP0064KB-A) 10 × 10 mm, 0.5 mm pitch<br>64-pin LQFP (PLQP0064GA-A) 14 × 14 mm, 0.8 mm pitch<br>64-pin WFLGA (PWLG0064KA-A) 5 × 5 mm, 0.5 mm pitch<br>48-pin LFQFP (PLQP0048KB-A) 7 × 7 mm, 0.5 mm pitch<br>48-pin HWQFN (PWQN0048KB-A) 7 × 7 mm, 0.5 mm pitch<br>40-pin HWQFN (PWQN0040KC-A) 6 × 6mm, 0.50mm pitch<br>36-pin WFLGA (PWLG0036KA-A) 4 × 4 mm, 0.5 mm pitch |
| On-chip debugging system | | E1 emulator (FINE interface) |

**Table 1.2 Comparison of Functions for Different Packages**

| Module/Functions | | RX111 Group | | | |
|---|---|---|---|---|---|
| | | 64 Pins | 48 Pins | 40 Pins | 36 Pins |
| Interrupts | External interrupts | NMI, IRQ0 to IRQ7 | | | |
| DMA | Data transfer controller | Supported | | | |
| Timers | Multi-function timer pulse unit 2 | 6 channels (MTU0 to MTU5) | | | |
| | Port output enable 2 | POE0# to POE3#, POE8# | | POE0#, POE2#, POE3#, POE8# | |
| | Compare match timer | 2 channels × 1 unit | | | |
| | Realtime clock | Supported | | Not supported | |
| | Independent watchdog timer | Supported | | | |
| Communication functions | Serial communications interfaces [simple $I^2C$, simple SPI] | 2 channels (SCI1, SCI5) | | | |
| | Serial communications interface [simple $I^2C$, simple SPI] | 1 channel (SCI12) | | | |
| | $I^2C$ bus interface | 1 channel | | | |
| | Serial peripheral interface | 1 channel | 1 channel (SSLA1 and SSLA3 are not supported) | 1 channel (SSLA1 to SSLA3 are not supported) | |
| | USB 2.0 host/function module (USBc) | 1 channel (Host/Function/ OTG) | 1 channel (Host/Function) | | |
| 12-bit A/D converter (including high-precision channels) | | 14 channels (6 channels) | 10 channels (4 channels) | 8 channels (3 channels) | 7 channels (2 channels) |
| D/A converter | | 2 channels | Not supported | | |
| Temperature sensor | | Supported | | | |
| CRC calculator | | Supported | | | |
| Event link controller | | Supported | | | |
| Packages | | 64-pin LFQFP 64-pin LQFP 64-pin WFLGA | 48-pin LFQFP 48-pin HWQFN | 40-pin HWQFN | 36-pin WFLGA |

## 1.2 List of Products

Table 1.3 is a list of products, and Figure 1.1 shows how to read the product part no., memory capacity, and package type.

**Table 1.3 List of Products (1/2)**

| Group | Part No. | Orderable Part No. | Package | ROM Capacity | RAM Capacity | E2 DataFlash | Maximum Operating Frequency | Operating Temperature |
|---|---|---|---|---|---|---|---|---|
| RX111 | R5F51115AGFM | R5F51115AGFM#30 | PLQP0064KB-A | 128 Kbytes | 16 Kbytes | 8 Kbytes | 32 MHz | –40 to +105°C |
| | R5F51115AGFK | R5F51115AGFK#30 | PLQP0064GA-A | | | | | |
| | R5F51115AGFL | R5F51115AGFL#30 | PLQP0048KB-A | | | | | |
| | R5F51115AGNE | R5F51115AGNE#V0 | PWQN0048KB-A | | | | | |
| | R5F51114AGFM | R5F51114AGFM#30 | PLQP0064KB-A | 96 Kbytes | | | | |
| | R5F51114AGFK | R5F51114AGFK#30 | PLQP0064GA-A | | | | | |
| | R5F51114AGFL | R5F51114AGFL#30 | PLQP0048KB-A | | | | | |
| | R5F51114AGNE | R5F51114AGNE#V0 | PWQN0048KB-A | | | | | |
| | R5F51113AGFM | R5F51113AGFM#30 | PLQP0064KB-A | 64 Kbytes | 10 Kbytes | | | |
| | R5F51113AGFK | R5F51113AGFK#30 | PLQP0064GA-A | | | | | |
| | R5F51113AGFL | R5F51113AGFL#30 | PLQP0048KB-A | | | | | |
| | R5F51113AGNE | R5F51113AGNE#V0 | PWQN0048KB-A | | | | | |
| | R5F51113AGNF | R5F51113AGNF#V0 | PWQN0040KC-A | | | | | |
| | R5F51111AGFM | R5F51111AGFM#30 | PLQP0064KB-A | 32 Kbytes | | | | |
| | R5F51111AGFK | R5F51111AGFK#30 | PLQP0064GA-A | | | | | |
| | R5F51111AGFL | R5F51111AGFL#30 | PLQP0048KB-A | | | | | |
| | R5F51111AGNE | R5F51111AGNE#V0 | PWQN0048KB-A | | | | | |
| | R5F51111AGNF | R5F51111AGNF#V0 | PWQN0040KC-A | | | | | |
| | R5F5111JAGFM | R5F5111JAGFM#30 | PLQP0064KB-A | 16 Kbytes | 8 Kbytes | | | |
| | R5F5111JAGFK | R5F5111JAGFK#30 | PLQP0064GA-A | | | | | |
| | R5F5111JAGFL | R5F5111JAGFL#30 | PLQP0048KB-A | | | | | |
| | R5F5111JAGNE | R5F5111JAGNE#V0 | PWQN0048KB-A | | | | | |
| | R5F5111JAGNF | R5F5111JAGNF#V0 | PWQN0040KC-A | | | | | |

**Table 1.3 List of Products (2/2)**

| Group | Part No. | Orderable Part No. | Package | ROM Capacity | RAM Capacity | E2 DataFlash | Maximum Operating Frequency | Operating Temperature |
|---|---|---|---|---|---|---|---|---|
| RX111 | R5F51115ADFM | R5F51115ADFM#30 | PLQP0064KB-A | 128 Kbytes | 16 Kbytes | 8 Kbytes | 32 MHz | –40 to +85°C |
| | R5F51115ADFK | R5F51115ADFK#30 | PLQP0064GA-A | | | | | |
| | R5F51115ADLF | R5F51115ADLF#U0 | PWLG0064KA-A | | | | | |
| | R5F51115ADFL | R5F51115ADFL#30 | PLQP0048KB-A | | | | | |
| | R5F51115ADNE | R5F51115ADNE#V0 | PVQN0048KA-A | | | | | |
| | R5F51114ADFM | R5F51114ADFM#30 | PLQP0064KB-A | 96 Kbytes | | | | |
| | R5F51114ADFK | R5F51114ADFK#30 | PLQP0064GA-A | | | | | |
| | R5F51114ADLF | R5F51114ADLF#U0 | PWLG0064KA-A | | | | | |
| | R5F51114ADFL | R5F51114ADFL#30 | PLQP0048KB-A | | | | | |
| | R5F51114ADNE | R5F51114ADNE#V0 | PWQN0048KB-A | | | | | |
| | R5F51113ADFM | R5F51113ADFM#30 | PLQP0064KB-A | 64 Kbytes | 10 Kbytes | | | |
| | R5F51113ADFK | R5F51113ADFK#30 | PLQP0064GA-A | | | | | |
| | R5F51113ADLF | R5F51113ADLF#U0 | PWLG0064KA-A | | | | | |
| | R5F51113ADFL | R5F51113ADFL#30 | PLQP0048KB-A | | | | | |
| | R5F51113ADNE | R5F51113ADNE#V0 | PWQN0048KB-A | | | | | |
| | R5F51113ADLM | R5F51113ADLM#U0 | PWLG0036KA-A | | | | | |
| | R5F51113ADNF | R5F51113ADNF#V0 | PWQN0040KC-A | | | | | |
| | R5F51111ADFM | R5F51111ADFM#30 | PLQP0064KB-A | 32 Kbytes | | | | |
| | R5F51111ADFK | R5F51111ADFK#30 | PLQP0064GA-A | | | | | |
| | R5F51111ADLF | R5F51111ADLF#U0 | PWLG0064KA-A | | | | | |
| | R5F51111ADFL | R5F51111ADFL#30 | PLQP0048KB-A | | | | | |
| | R5F51111ADNE | R5F51111ADNE#V0 | PWQN0048KB-A | | | | | |
| | R5F51111ADLM | R5F51111ADLM#U0 | PWLG0036KA-A | | | | | |
| | R5F51111ADNF | R5F51111ADNF#V0 | PWQN0040KC-A | | | | | |
| | R5F5111JADFM | R5F5111JADFM#30 | PLQP0064KB-A | 16 Kbytes | 8 Kbytes | | | |
| | R5F5111JADFK | R5F5111JADFK#30 | PLQP0064GA-A | | | | | |
| | R5F5111JADLF | R5F5111JADLF#U0 | PWLG0064KA-A | | | | | |
| | R5F5111JADFL | R5F5111JADFL#30 | PLQP0048KB-A | | | | | |
| | R5F5111JADNE | R5F5111JADNE#V0 | PWQN0048KB-A | | | | | |
| | R5F5111JADLM | R5F5111JADLM#U0 | PWLG0036KA-A | | | | | |
| | R5F5111JADNF | R5F5111JADNF#V0 | PWQN0040KC-A | | | | | |

Note: • Orderable part numbers are current as of when this manual was published. Please make sure to refer the relevant product page on the Renesas website for the latest part numbers.

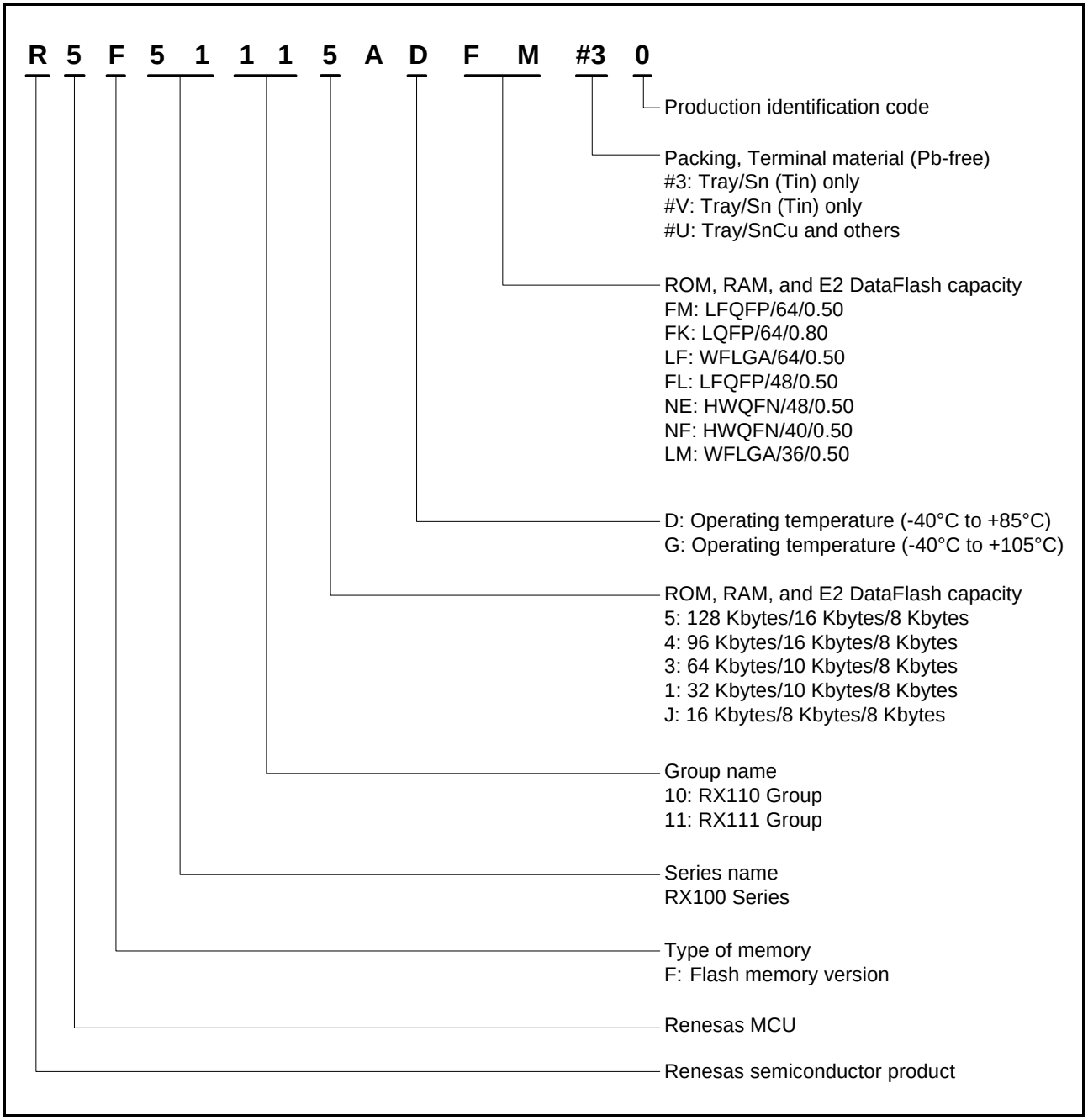
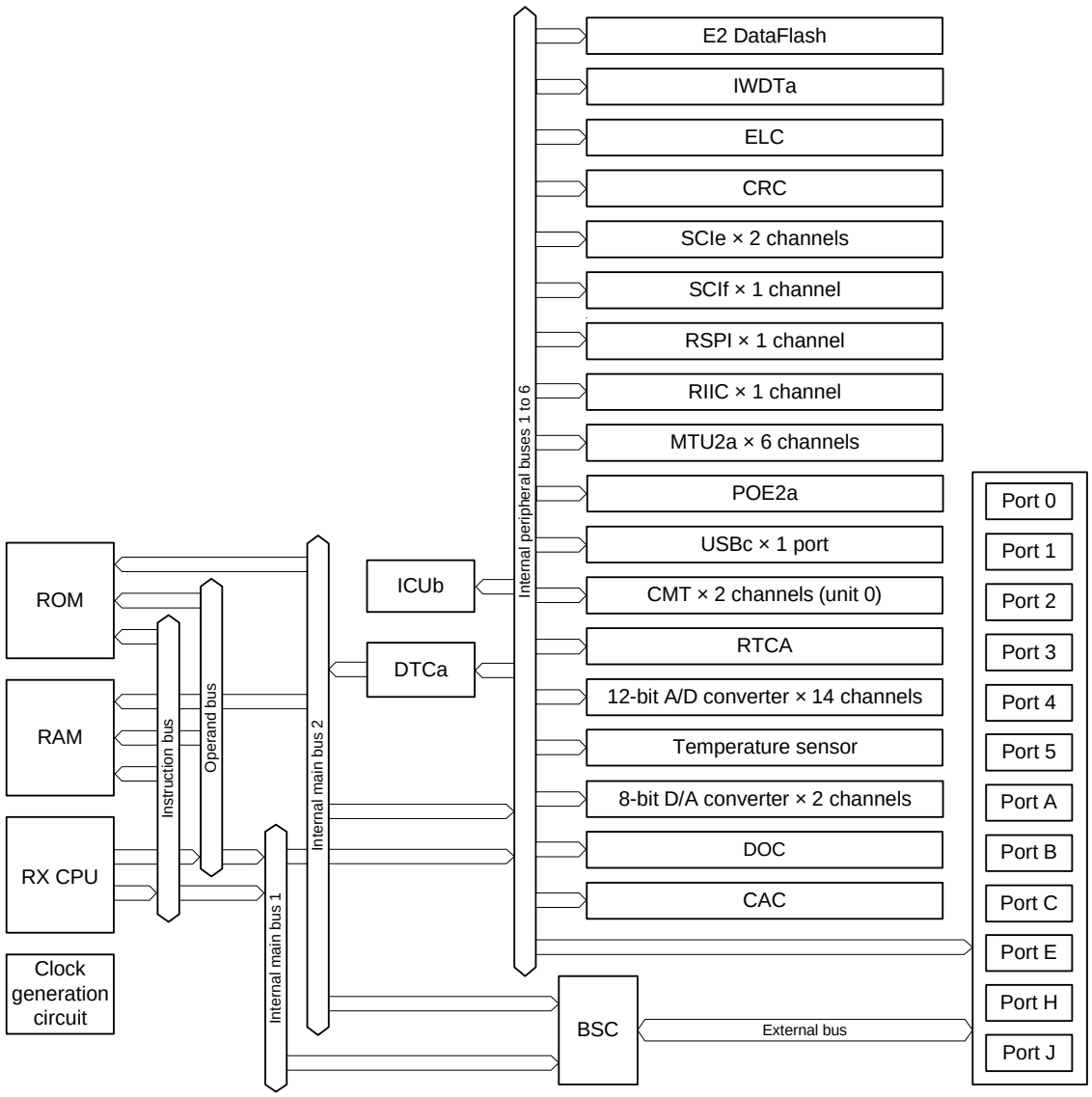



**Figure 1.1 How to Read the Product Part No., Memory Capacity, and Package Type**

## 1.3 Block Diagram

Figure 1.2 shows a block diagram.




ICUb: Interrupt controller
DTCa: Data transfer controller
IWDTa: Independent watchdog timer
ELC: Event link controller
CRC: CRC (cyclic redundancy check) calculator
SCIe/SCIf: Serial communications interface
RSPI: Serial peripheral interface
RIIC: $I^2C$ bus interface
MTU2a: Multi-function timer pulse unit 2
POE2a: Port output enable 2
USBc: USB 2.0 host/function module
CMT: Compare match timer
RTCA: Realtime clock
DOC: Data operation circuit
CAC: Clock frequency accuracy measurement circuit

**Figure 1.2 Block Diagram**

## 1.4 Pin Functions

Table 1.4 lists the pin functions.

**Table 1.4 Pin Functions (1/3)**

| Classifications | Pin Name | I/O | Description |
|---|---|---|---|
| Power supply | VCC | Input | Power supply pin. Connect it to the system power supply. |
| | VCL | — | Connect this pin to the VSS pin via the 4.7 µF smoothing capacitor used to stabilize the internal power supply. Place the capacitor close to the pin. |
| | VSS | Input | Ground pin. Connect it to the system power supply (0 V). |
| Clock | XTAL | Output/ Input *1 | Pins for connecting a crystal. An external clock can be input through the XTAL pin. |
| | EXTAL | Input | |
| | XCIN | Input | Input/output pins for the sub-clock oscillator. Connect a crystal between XCIN and XCOUT. |
| | XCOUT | Output | |
| | CLKOUT | Output | Clock output pin |
| Operating mode control | MD | Input | Pin for setting the operating mode. The signal levels on this pin must not be changed during operation. |
| | UB# | Input | Pin used for USB interface mode of boot mode. |
| | UPSEL | Input | Pin used for USB interface mode of boot mode. |
| System control | RES# | Input | Reset pin. This MCU enters the reset state when this signal goes low. |
| CAC | CACREF | Input | Input pin for the clock frequency accuracy measurement circuit. |
| On-chip emulator | FINED | I/O | FINE interface pin. |
| LVD | CMPA2 | Input | Detection target voltage pin for voltage detection 2 |
| Interrupts | NMI | Input | Non-maskable interrupt request pin. |
| | IRQ0 to IRQ7 | Input | Interrupt request pins. |
| Multi-function timer pulse unit 2 | MTIOC0A, MTIOC0B MTIOC0C, MTIOC0D | I/O | The TGRA0 to TGRD0 input capture input/output compare output/PWM output pins. |
| | MTIOC1A, MTIOC1B | I/O | The TGRA1 and TGRB1 input capture input/output compare output/PWM output pins. |
| | MTIOC2A, MTIOC2B | I/O | The TGRA2 and TGRB2 input capture input/output compare output/PWM output pins. |
| | MTIOC3A, MTIOC3B MTIOC3C, MTIOC3D | I/O | The TGRA3 to TGRD3 input capture input/output compare output/PWM output pins. |
| | MTIOC4A, MTIOC4B MTIOC4C, MTIOC4D | I/O | The TGRA4 to TGRD4 input capture input/output compare output/PWM output pins. |
| | MTIC5U, MTIC5V, MTIC5W | Input | The TGRU5, TGRV5, and TGRW5 input capture input/external pulse input pins. |
| | MTCLKA, MTCLKB, MTCLKC, MTCLKD | Input | Input pins for the external clock. |
| Port output enable 2 | POE0# to POE3#, POE8# | Input | Input pins for request signals to place the MTU pins in the high impedance state. |
| Realtime clock | RTCOUT | Output | Output pin for the 1-Hz/64-Hz clock. |
| Serial communications interface (SCIe) | • Asynchronous mode/clock synchronous mode | | |
| | SCK1, SCK5 | I/O | Input/output pins for the clock. |
| | RXD1, RXD5 | Input | Input pins for received data. |
| | TXD1, TXD5 | Output | Output pins for transmitted data. |
| | CTS1#, CTS5# | Input | Input pins for controlling the start of transmission and reception. |
| | RTS1#, RTS5# | Output | Output pins for controlling the start of transmission and reception. |

**Table 1.4 Pin Functions (2/3)**

| Classifications | Pin Name | I/O | Description |
|---|---|---|---|
| Serial communications interface (SCIe) | • Simple I2C mode | | |
| | SSCL1, SSCL5 | I/O | Input/output pins for the I2C clock. |
| | SSDA1, SSDA5 | I/O | Input/output pins for the I2C data. |
| | • Simple SPI mode | | |
| | SCK1, SCK5 | I/O | Input/output pins for the clock. |
| | SMISO1, SMISO5 | I/O | Input/output pins for slave transmit data. |
| | SMOSI1, SMOSI5 | I/O | Input/output pins for master transmit data. |
| | SS1#, SS5# | Input | Chip-select input pins. |
| Serial communications interface (SCIf) | • Asynchronous mode/clock synchronous mode | | |
| | SCK12 | I/O | Input/output pin for the clock. |
| | RXD12 | Input | Input pin for receiving data. |
| | TXD12 | Output | Output pin for transmitting data. |
| | CTS12# | Input | Input pin for controlling the start of transmission and reception. |
| | RTS12# | Output | Output pin for controlling the start of transmission and reception. |
| | • Simple I2C mode | | |
| | SSCL12 | I/O | Input/output pin for the I2C clock. |
| | SSDA12 | I/O | Input/output pin for the I2C data. |
| | • Simple SPI mode | | |
| | SCK12 | I/O | Input/output pin for the clock. |
| | SMISO12 | I/O | Input/output pin for slave transmit data. |
| | SMOSI12 | I/O | Input/output pin for master transmit data. |
| | SS12# | Input | Chip-select input pin. |
| | • Extended serial mode | | |
| | RXDX12 | Input | Input pin for data reception by SCIf. |
| | TXDX12 | Output | Output pin for data transmission by SCIf. |
| | SIOX12 | I/O | Input/output pin for data reception or transmission by SCIf. |
| I2C bus interface | SCL0 | I/O | Input/output pin for I2C bus interface clocks. Bus can be directly driven by the N-channel open drain output. |
| | SDA0 | I/O | Input/output pin for I2C bus interface data. Bus can be directly driven by the N-channel open drain output. |
| Serial peripheral interface | RSPCKA | I/O | Input/output pin for the RSPI clock. |
| | MOSIA | I/O | Input/output pin for transmitting data from the RSPI master. |
| | MISOA | I/O | Input/output pin for transmitting data from the RSPI slave. |
| | SSLA0 | I/O | Input/output pin to select the slave for the RSPI. |
| | SSLA1 to SSLA3 | Output | Output pins to select the slave for the RSPI. |
| USB 2.0 host/ function module | VCC_USB | Input | Power supply pin for USB. Connect this pin to VCC. |
| | VSS_USB | Input | Ground pin for USB. Connect this pin to VSS. |
| | USB0_DP | I/O | D+ I/O pin of the USB on-chip transceiver. |
| | USB0_DM | I/O | D- I/O pin of the USB on-chip transceiver. |
| | USB0_VBUS | Input | USB cable connection monitor pin. |
| | USB0_EXICEN | Output | Low-power control signal for the OTG chip. |
| | USB0_VBUSEN | Output | VBUS (5 V) supply enable signal for the OTG chip. |
| | USB0_OVRCURA, USB0_OVRCURB | Input | External overcurrent detection pins. |
| | USB0_ID | Input | Mini-AB connector ID input pin during operation in OTG mode. |

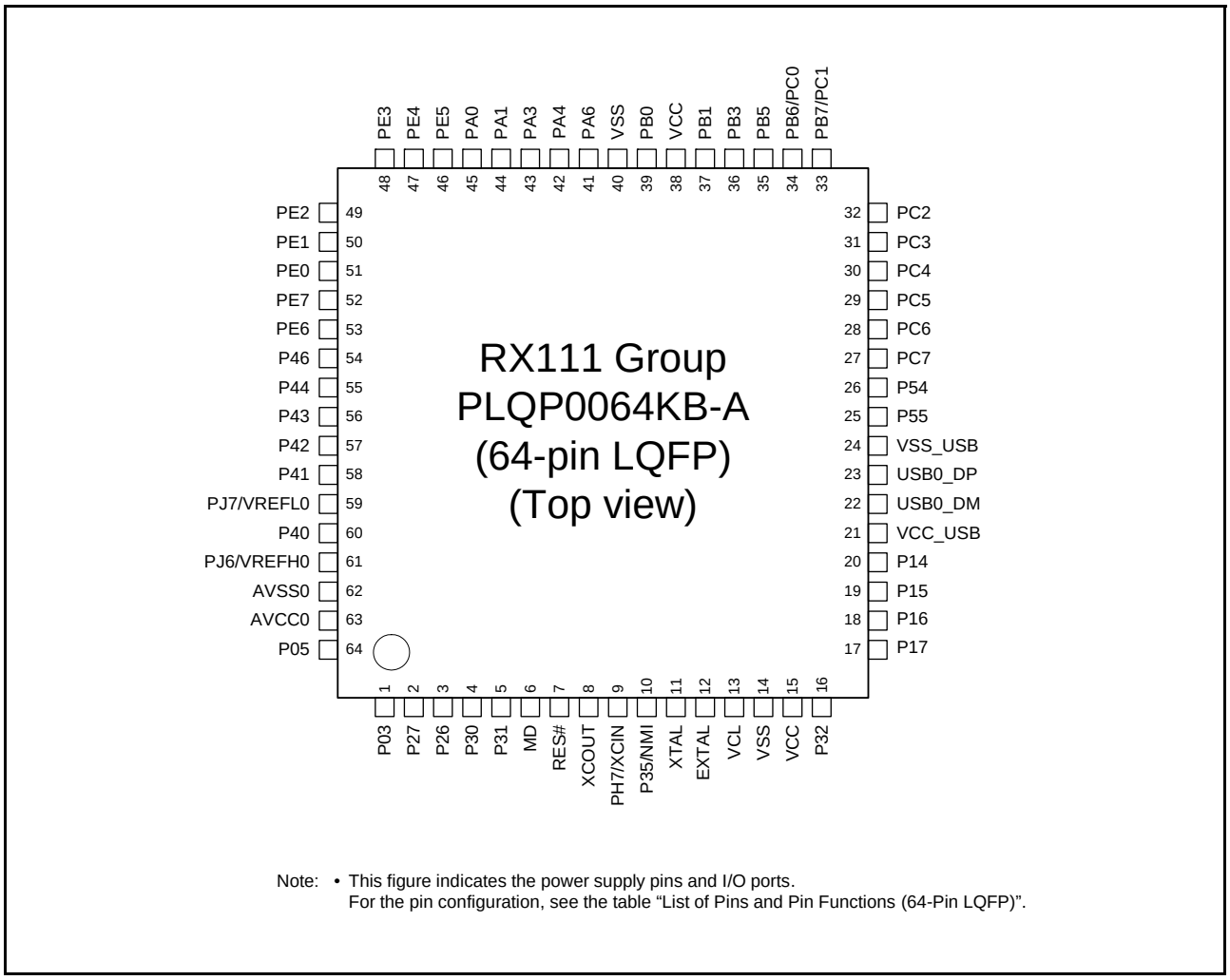
**Table 1.4 Pin Functions (3/3)**

| Classifications | Pin Name | I/O | Description |
|---|---|---|---|
| 12-bit A/D converter | AN000 to AN004, AN006, AN008 to AN015 | Input | Input pins for the analog signals to be processed by the A/D converter. |
| | ADTRG0# | Input | Input pin for the external trigger signals that start the A/D conversion. |
| D/A converter | DA0, DA1 | Output | Output pins for the analog signals to be processed by the D/A converter. |
| Analog power supply | AVCC0 | Input | Analog voltage supply pin for the 12-bit A/D converter. Connect this pin to VCC when not using the 12-bit A/D converter. |
| | AVSS0 | Input | Analog ground pin for the 12-bit A/D converter. Connect this pin to VSS when not using the 12-bit A/D converter. |
| | VREFH0 | Input | Analog reference voltage supply pin for the 12-bit A/D converter. Connect this pin to VCC when not using the 12-bit A/D converter. |
| | VREFL0 | Input | Analog reference ground pin for the 12-bit A/D converter. Connect this pin to VSS when not using the 12-bit A/D converter. |
| I/O ports | P03, P05 | I/O | 2-bit input/output pins. |
| | P14 to P17 | I/O | 4-bit input/output pins. |
| | P26, P27 | I/O | 2-bit input/output pins. |
| | P30 to P32, P35 | I/O | 4-bit input/output pins (P35 input pin). |
| | P40 to P44, P46 | I/O | 6-bit input/output pins. |
| | P54, P55 | I/O | 2-bit input/output pins. |
| | PA0, PA1, PA3, PA4, PA6 | I/O | 5-bit input/output pins. |
| | PB0, PB1, PB3, PB5 to PB7 | I/O | 6-bit input/output pins. |
| | PC0 to PC7 | I/O | 8-bit input/output pins. |
| | PE0 to PE7 | I/O | 8-bit input/output pins. |
| | PH7 | Input | 1-bit input pin. |
| | PJ6, PJ7 | I/O | 2-bit input/output pins. |

Note 1. For external clock input.

## 1.5 Pin Assignments

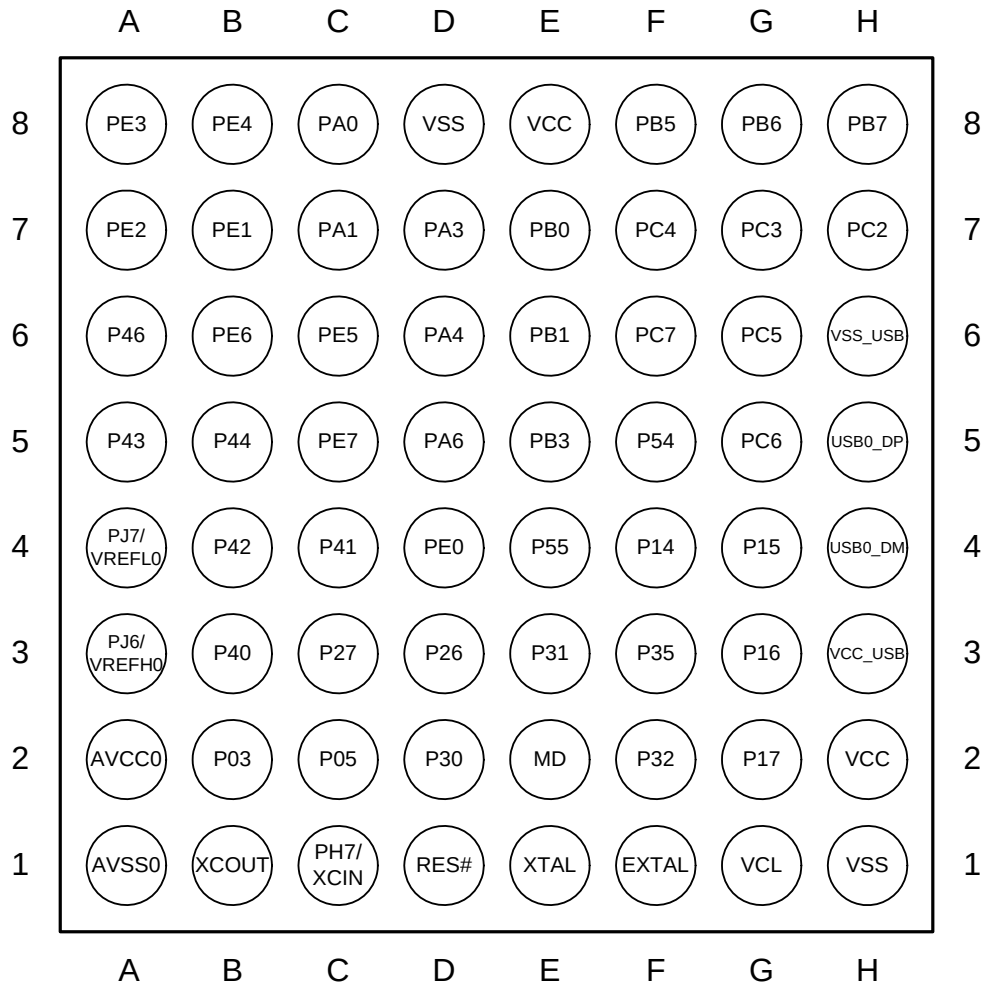
Figure 1.3 to Figure 1.7 show the pin assignments. Table 1.5 to Table 1.6 show the lists of pins and pin functions.




Note: • This figure indicates the power supply pins and I/O ports.
For the pin configuration, see the table “List of Pins and Pin Functions (64-Pin LQFP)”.

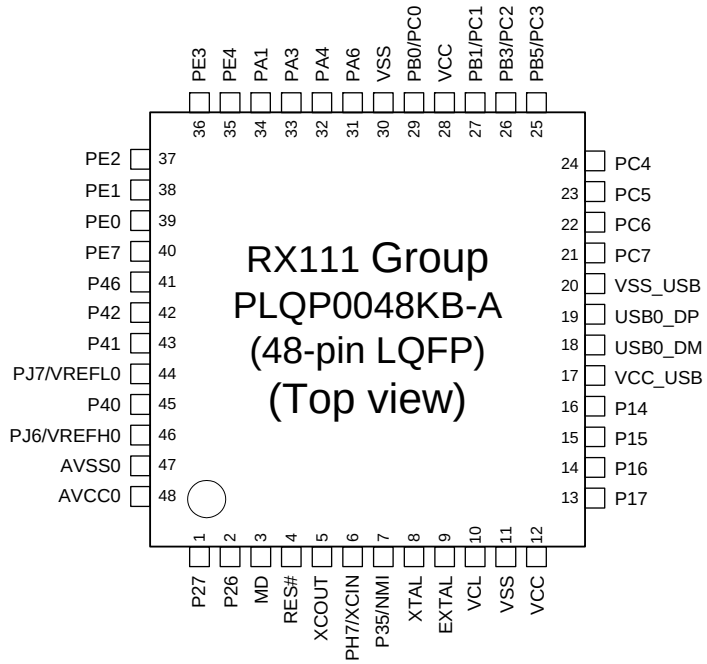
**Figure 1.3 Pin Assignments of the 64-Pin LQFP**

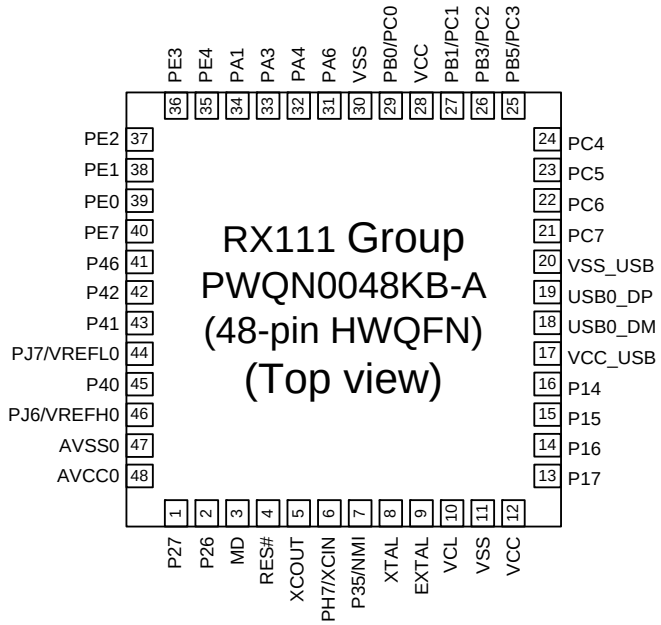
RX111 Group
PWLG0064KA-A
(64-pin WFLGA)
(Upper perspective view)




Note: • This figure indicates the power supply pins and I/O port pins. For the pin configuration, see the table “List of Pins and Pin Functions (64-Pin WFLGA)”.
• For the position of A1 pin in the package, see “Package Dimensions”.

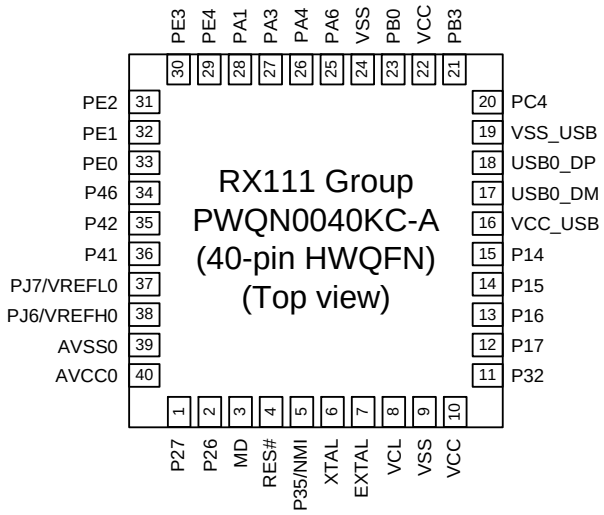
**Figure 1.4 Pin Assignments of the 64-Pin WFLGA**




Note: • This figure indicates The power supply pins and I/O port pins.
For the pin configuration, see the table “List of Pins and Pin Functions (48-Pin LQFP/HWQFN)”.

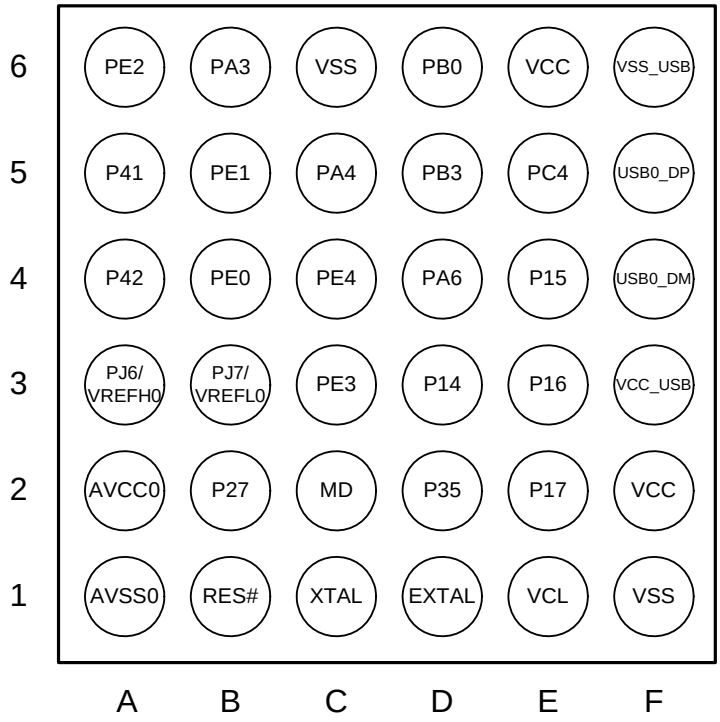
**Figure 1.5 Pin Assignments of the 48-Pin LQFP/HWQFN**




Note: • This figure indicates the power supply pins and I/O port pins.
For the pin configuration, see the table “List of Pins and Pin Functions (40-Pin HWQFN)”.

**Figure 1.6 Pin Assignments of the 40-Pin HWQFN**

RX111 Group
PWLG0036KA-A
(36-pin WFLGA)
(Upper perspective view)




Note: • This figure indicates the power supply pins and I/O port pins. For the pin configuration, see the table “List of Pins and Pin Functions (36-Pin WFLGA)”.
• For the position of A1 pin in the package, see “Package Dimensions”.

**Figure 1.7 Pin Assignments of the 36-Pin WFLGA**

**Table 1.5 List of Pins and Pin Functions (64-Pin LQFP) (1/2)**

| Pin No. | Power Supply, Clock, System Control | I/O Port | Timers (MTU, POE, RTC) | Communication (SCIe, SCIf, RSPI, RIIC, USB) | Others |
|---|---|---|---|---|---|
| 1 | | P03 | | | DA0 |
| 2 | | P27 | MTIOC2B | SCK1/SCK12 | IRQ3/CMPA2/ CACREF/ADTRG0# |
| 3 | | P26 | MTIOC2A | TXD1/SMOSI1/SSDA1/USB0_VBUSEN | |
| 4 | | P30 | MTIOC4B/POE8# | RXD1/SMISO1/SSCL1 | IRQ0 |
| 5 | | P31 | MTIOC4D | CTS1#/RTS1#/SS1# | IRQ1 |
| 6 | MD | | | | FINED |
| 7 | RES# | | | | |
| 8 | XCOUT | | | | |
| 9 | XCIN | PH7 | | | |
| 10 | UPSEL | P35 | | | NMI |
| 11 | XTAL | | | | |
| 12 | EXTAL | | | | |
| 13 | VCL | | | | |
| 14 | VSS | | | | |
| 15 | VCC | | | | |
| 16 | | P32 | MTIOC0C/RTCOUT | | IRQ2 |
| 17 | | P17 | MTIOC0C/MTIOC3A/ MTIOC3B/POE8# | SCK1/MISOA/SDA0/RXD12/RXDX12/ SMISO12/SSCL12 | IRQ7 |
| 18 | | P16 | MTIOC3C/MTIOC3D/ RTCOUT | TXD1/SMOSI1/SSDA1/MOSIA/SCL0/ USB0_VBUS/USB0_VBUSEN/ USB0_OVRCURB | IRQ6/ADTRG0# |
| 19 | | P15 | MTIOC0B/MTCLKB | RXD1/SMISO1/SSCL1/RSPCKA | IRQ5/CLKOUT |
| 20 | UB# | P14 | MTIOC0A/MTIOC3A/ MTCLKA | CTS1#/RTS1#/SS1#/SSLA0/TXD12/ TXDX12/SIOX12/SMOSI12/SSDA12/ USB0_OVRCURA | IRQ4 |
| 21 | VCC_USB | | | | |
| 22 | | | | USB0_DM | |
| 23 | | | | USB0_DP | |
| 24 | VSS_USB | | | | |
| 25 | | P55 | MTIOC4D | | |
| 26 | | P54 | MTIOC4B | | |
| 27 | | PC7 | MTIOC3A/MTCLKB | TXD1/SMOSI1/SSDA1/MISOA/ USB0_OVRCURB | CACREF |
| 28 | | PC6 | MTIOC3C/MTCLKA | RXD1/SMISO1/SSCL1/MOSIA/ USB0_EXICEN | |
| 29 | | PC5 | MTIOC3B/MTCLKD | SCK1/RSPCKA/USB0_ID | |
| 30 | | PC4 | MTIOC3D/MTCLKC/POE0# | SCK5/SSLA0/USB0_VBUS*1/ USB0_VBUSEN | IRQ2/CLKOUT |
| 31 | | PC3 | MTIOC4D | TXD5/SMOSI5/SSDA5 | |
| 32 | | PC2 | MTIOC4B | RXD5/SMISO5/SSCL5/SSLA3 | |
| 33 | | PB7/PC1 | MTIOC3B | | |
| 34 | | PB6/PC0 | MTIOC3D | | |
| 35 | | PB5 | MTIOC2A/MTIOC1B/POE1# | | |
| 36 | | PB3 | MTIOC0A/MTIOC3B/ MTIOC4A/POE3# | USB0_OVRCURA | |
| 37 | | PB1 | MTIOC0C/MTIOC4C | | IRQ4 |
| 38 | VCC | | | | |
| 39 | | PB0 | MTIC5W/MTIOC0C/ RTCOUT | SCL0/RSPCKA | IRQ2/ADTRG0# |
| 40 | VSS | | | | |
| 41 | | PA6 | MTIC5V/MTCLKB/MTIOC2A/ POE2# | CTS5#/RTS5#/SS5#/SDA0/MOSIA | IRQ3 |

**Table 1.5 List of Pins and Pin Functions (64-Pin LQFP) (2/2)**

| Pin No. | Power Supply, Clock, System Control | I/O Port | Timers (MTU, POE, RTC) | Communication (SCIe, SCIf, RSPI, RIIC, USB) | Others |
|---|---|---|---|---|---|
| 42 | | PA4 | MTIC5U/MTCLKA/MTIOC2B | TXD5/SMOSI5/SSDA5/SSLA0 | IRQ5 |
| 43 | | PA3 | MTIOC0D/MTCLKD/ MTIOC1B/POE0# | RXD5/SMISO5/SSCL5/MISOA | IRQ6 |
| 44 | | PA1 | MTIOC0B/MTCLKC/ RTCOUT | SCK5/SSLA2 | |
| 45 | | PA0 | MTIOC4A | SSLA1 | CACREF |
| 46 | | PE5 | MTIOC4C/MTIOC2B | | IRQ5/AN013 |
| 47 | | PE4 | MTIOC4D/MTIOC1A/ MTIOC3A | MOSIA | IRQ4/AN012 |
| 48 | | PE3 | MTIOC0A/MTIOC1B/ MTIOC4B/POE8# | CTS12#/RTS12#/SS12#/RSPCKA | IRQ3/AN011 |
| 49 | | PE2 | MTIOC4A | RXD12/RXDX12/SMISO12/SSCL12 | IRQ7/AN010 |
| 50 | | PE1 | MTIOC4C | TXD12/TXDX12/SIOX12/SMOSI12/ SSDA12 | IRQ1/AN009 |
| 51 | | PE0 | MTIOC2A/POE3# | SCK12 | IRQ0/AN008 |
| 52 | | PE7 | | | IRQ7/AN015 |
| 53 | | PE6 | | | IRQ6/AN014 |
| 54 | | P46 | | | AN006 |
| 55 | | P44 | | | AN004 |
| 56 | | P43 | | | AN003 |
| 57 | | P42 | | | AN002 |
| 58 | | P41 | | | AN001 |
| 59 | VREFL0 | PJ7 | | | |
| 60 | | P40 | | | AN000 |
| 61 | VREFH0 | PJ6 | | | |
| 62 | AVSS0 | | | | |
| 63 | AVCC0 | | | | |
| 64 | | P05 | | | DA1 |

Note 1. Not 5 V tolerant.

**Table 1.6 List of Pins and Pin Functions (64-Pin WFLGA) (1/2)**

| Pin No. | Power Supply, Clock, System Control | I/O Port | Timers (MTU, POE, RTC) | Communication (SCIe, SCIf, RSPI, RIIC, USB) | Others |
|---|---|---|---|---|---|
| A1 | AVSS0 | | | | |
| A2 | AVCC0 | | | | |
| A3 | VREFH0 | PJ6 | | | |
| A4 | VREFL0 | PJ7 | | | |
| A5 | | P43 | | | AN003 |
| A6 | | P46 | | | AN006 |
| A7 | | PE2 | MTIOC4A | RXD12/RXDX12/SMISO12/SSCL12 | IRQ7/AN010 |
| A8 | | PE3 | MTIOC0A/MTIOC1B/ MTIOC4B/POE8# | CTS12#/RTS12#/SS12#/RSPCKA | IRQ3/AN011 |
| B1 | XCOUT | | | | |
| B2 | | P03 | | | DA0 |
| B3 | | P40 | | | AN000 |
| B4 | | P42 | | | AN002 |
| B5 | | P44 | | | AN004 |
| B6 | | PE6 | | | IRQ6/AN014 |
| B7 | | PE1 | MTIOC4C | TXD12/TXDX12/SIOX12/SMOSI12/ SSDA12 | IRQ1/AN009 |
| B8 | | PE4 | MTIOC1A/MTIOC3A/ MTIOC4D | MOSIA | IRQ4/AN012 |
| C1 | XCIN | PH7 | | | |
| C2 | | P05 | | | DA1 |
| C3 | | P27 | MTIOC2B | SCK1/SCK12 | IRQ3/CMPA2/CACREF/ ADTRG0# |
| C4 | | P41 | | | AN001 |
| C5 | | PE7 | | | IRQ7/AN015 |
| C6 | | PE5 | MTIOC2B/MTIOC4C | | IRQ5/AN013 |
| C7 | | PA1 | MTIOC0B/MTCLKC/ RTCOUT | SCK5/SSLA2 | |
| C8 | | PA0 | MTIOC4A | SSLA1 | CACREF |
| D1 | RES# | | | | |
| D2 | | P30 | MTIOC4B/POE8# | RXD1/SMISO1/SSCL1 | IRQ0 |
| D3 | | P26 | MTIOC2A | TXD1/SMOSI1/SSDA1/ USB0_VBUSEN | |
| D4 | | PE0 | MTIOC2A/POE3# | SCK12 | IRQ0/AN008 |
| D5 | | PA6 | MTIC5V/MTIOC2A/MTCLKB/ POE2# | CTS5#/RTS5#/SS5#/SDA0/MOSIA | IRQ3 |
| D6 | | PA4 | MTIC5U/MTIOC2B/MTCLKA | TXD5/SMOSI5/SSDA5/SSLA0 | IRQ5 |
| D7 | | PA3 | MTIOC0D/MTCLKD/ MTIOC1B/POE0# | RXD5/SMISO5/SSCL5/MISOA | IRQ6 |
| D8 | VSS | | | | |
| E1 | XTAL | | | | |
| E2 | MD | | | | FINED |
| E3 | | P31 | MTIOC4D | CTS1#/RTS1#/SS1# | IRQ1 |
| E4 | | P55 | MTIOC4D | | |
| E5 | | PB3 | MTIOC0A/MTIOC3B/ MTIOC4A/POE3# | USB0_OVRCURA | |
| E6 | | PB1 | MTIOC0C/MTIOC4C | | IRQ4 |
| E7 | | PB0 | MTIC5W/MTIOC0C/ RTCOUT | SCL0/RSPCKA | IRQ2/ADTRG0# |
| E8 | VCC | | | | |
| F1 | EXTAL | | | | |

**Table 1.6 List of Pins and Pin Functions (64-Pin WFLGA) (2/2)**

| Pin No. | Power Supply, Clock, System Control | I/O Port | Timers (MTU, POE, RTC) | Communication (SCIe, SCIf, RSPI, RIIC, USB) | Others |
|---|---|---|---|---|---|
| F2 | | P32 | MTIOC0C/RTCOUT | | IRQ2 |
| F3 | UPSEL | P35 | | | NMI |
| F4 | UB# | P14 | MTIOC0A/MTIOC3A/ MTCLKA | CTS1#/RTS1#/SS1#/TXD12/ TXDX12/SIOX12/SMOSI12/ SSDA12/SSLA0/USB0_OVRCURA | IRQ4 |
| F5 | | P54 | MTIOC4B | | |
| F6 | | PC7 | MTIOC3A/MTCLKB | TXD1/SMOSI1/SSDA1/MISOA/ USB0_OVRCURB | CACREF |
| F7 | | PC4 | MTCLKC/MTIOC3D/POE0# | SCK5/SSLA0/USB0_VBUSEN/ USB0_VBUS*1 | IRQ2/CLKOUT |
| F8 | | PB5 | MTIOC1B/MTIOC2A/POE1# | | |
| G1 | VCL | | | | |
| G2 | | P17 | MTIOC0C/MTIOC3A/ MTIOC3B/POE8# | SCK1/MISOA/SDA0/RXD12/ RXDX12/SMISO12/SSCL12 | IRQ7 |
| G3 | | P16 | MTIOC3C/MTIOC3D/ RTCOUT | TXD1/SMOSI1/SSDA1/SCL0/ MOSIA/USB0_VBUSEN/ USB0_OVRCURB/USB0_VBUS | IRQ6/ADTRG0# |
| G4 | | P15 | MTIOC0B/MTCLKB | RXD1/SMISO1/SSCL1/RSPCKA | IRQ5/CLKOUT |
| G5 | | PC6 | MTIOC3C/MTCLKA | RXD1/SMISO1/SSCL1/MOSIA/ USB0_EXICEN | |
| G6 | | PC5 | MTIOC3B/MTCLKD | SCK1/RSPCKA/USB0_ID | |
| G7 | | PC3 | MTIOC4D | TXD5/SMOSI5/SSDA5 | |
| G8 | | PB6/PC0 | MTIOC3D | | |
| H1 | VSS | | | | |
| H2 | VCC | | | | |
| H3 | VCC_USB | | | | |
| H4 | | | | USB0_DM | |
| H5 | | | | USB0_DP | |
| H6 | VSS_USB | | | | |
| H7 | | PC2 | MTIOC4B | RXD5/SMISO5/SSCL5/SSLA3 | |
| H8 | | PB7/PC1 | MTIOC3B | | |

Note 1. Not 5 V tolerant.

**Table 1.7 List of Pins and Pin Functions (48-Pin LQFP/HWQFN) (1/2)**

| Pin No. | Power Supply, Clock, System Control | I/O Port | Timers (MTU, POE, RTC) | Communication (SCIe, SCIf, RSPI, RIIC, USB) | Others |
|---|---|---|---|---|---|
| 1 | | P27 | MTIOC2B | SCK1/SCK12 | IRQ3/CMPA2/ CACREF/ADTRG0# |
| 2 | | P26 | MTIOC2A | TXD1/SMOSI1/SSDA1/USB0_VBUSEN | |
| 3 | MD | | | | FINED |
| 4 | RES# | | | | |
| 5 | XCOUT | | | | |
| 6 | XCIN | PH7 | | | |
| 7 | UPSEL | P35 | | | NMI |
| 8 | XTAL | | | | |
| 9 | EXTAL | | | | |
| 10 | VCL | | | | |
| 11 | VSS | | | | |
| 12 | VCC | | | | |
| 13 | | P17 | MTIOC0C/MTIOC3A/ MTIOC3B/POE8# | SCK1/MISOA/SDA0/RXD12/RXDX12/ SMISO12/SSCL12 | IRQ7 |
| 14 | | P16 | MTIOC3C/MTIOC3D/ RTCOUT | TXD1/SMOSI1/SSDA1/MOSIA/SCL0/ USB0_VBUS/USB0_VBUSEN/ USB0_OVRCURB | IRQ6/ADTRG0# |
| 15 | | P15 | MTIOC0B/MTCLKB | RXD1/SMISO1/SSCL1/RSPCKA | IRQ5/CLKOUT |
| 16 | UB# | P14 | MTIOC0A/MTIOC3A/ MTCLKA | CTS1#/RTS1#/SS1#/SSLA0/TXD12/ TXDX12/SIOX12/SMOSI12/SSDA12/ USB0_OVRCURA | IRQ4 |
| 17 | VCC_USB | | | | |
| 18 | | | | USB0_DM | |
| 19 | | | | USB0_DP | |
| 20 | VSS_USB | | | | |
| 21 | | PC7 | MTIOC3A/MTCLKB | TXD1/SMOSI1/SSDA1/MISOA/ USB0_OVRCURB | CACREF |
| 22 | | PC6 | MTIOC3C/MTCLKA | RXD1/SMISO1/SSCL1/MOSIA/ USB0_EXICEN | |
| 23 | | PC5 | MTIOC3B/MTCLKD | SCK1/RSPCKA/USB0_ID | |
| 24 | | PC4 | MTIOC3D/MTCLKC/POE0# | SCK5/SSLA0/USB0_VBUS*1/ USB0_VBUSEN | IRQ2/CLKOUT |
| 25 | | PB5/PC3 | MTIOC2A/MTIOC1B/POE1# | | |
| 26 | | PB3/PC2 | MTIOC0A/MTIOC3B/ MTIOC4A/POE3# | USB0_OVRCURA | |
| 27 | | PB1/PC1 | MTIOC0C/MTIOC4C | | IRQ4 |
| 28 | VCC | | | | |
| 29 | | PB0/PC0 | MTIC5W/MTIOC0C/ RTCOUT | SCL0/RSPCKA | IRQ2/ADTRG0# |
| 30 | VSS | | | | |
| 31 | | PA6 | MTIC5V/MTCLKB/MTIOC2A/ POE2# | CTS5#/RTS5#/SS5#/SDA0/MOSIA | IRQ3 |
| 32 | | PA4 | MTIC5U/MTCLKA/MTIOC2B | TXD5/SMOSI5/SSDA5/SSLA0 | IRQ5 |
| 33 | | PA3 | MTIOC0D/MTCLKD/ MTIOC1B/POE0# | RXD5/SMISO5/SSCL5/MISOA | IRQ6 |
| 34 | | PA1 | MTIOC0B/MTCLKC/ RTCOUT | SCK5/SSLA2 | |
| 35 | | PE4 | MTIOC4D/MTIOC1A/ MTIOC3A | MOSIA | IRQ4/AN012 |
| 36 | | PE3 | MTIOC0A/MTIOC1B/ MTIOC4B/POE8# | CTS12#/RTS12#/SS12#/RSPCKA | IRQ3/AN011 |
| 37 | | PE2 | MTIOC4A | RXD12/RXDX12/SMISO12/SSCL12 | IRQ7/AN010 |
| 38 | | PE1 | MTIOC4C | TXD12/TXDX12/SIOX12/SMOSI12/ SSDA12 | IRQ1/AN009 |

**Table 1.7 List of Pins and Pin Functions (48-Pin LQFP/HWQFN) (2/2)**

| Pin No. | Power Supply, Clock, System Control | I/O Port | Timers (MTU, POE, RTC) | Communication (SCIe, SCIf, RSPI, RIIC, USB) | Others |
|---|---|---|---|---|---|
| 39 | | PE0 | MTIOC2A/POE3# | SCK12 | IRQ0/AN008 |
| 40 | | PE7 | | | IRQ7/AN015 |
| 41 | | P46 | | | AN006 |
| 42 | | P42 | | | AN002 |
| 43 | | P41 | | | AN001 |
| 44 | VREFL0 | PJ7 | | | |
| 45 | | P40 | | | AN000 |
| 46 | VREFH0 | PJ6 | | | |
| 47 | AVSS0 | | | | |
| 48 | AVCC0 | | | | |

Note 1. Not 5 V tolerant.

**Table 1.8 List of Pins and Pin Functions (40-Pin HWQFN) (1/2)**

| Pin No. | Power Supply, Clock, System Control | I/O Port | Timers (MTU, POE, RTC) | Communication (SCIe, SCIf, RSPI, RIIC, USB) | Others |
|---|---|---|---|---|---|
| 1 | | P27 | MTIOC2B | SCK1/SCK12 | IRQ3/CMPA2/CACREF/ ADTRG0# |
| 2 | | P26 | MTIOC2A | TXD1/SMOSI1/SSDA1/ USB0_VBUSEN | |
| 3 | MD | | | | FINED |
| 4 | RES# | | | | |
| 5 | UPSEL | P35 | | | NMI |
| 6 | XTAL | | | | |
| 7 | EXTAL | | | | |
| 8 | VCL | | | | |
| 9 | VSS | | | | |
| 10 | VCC | | | | |
| 11 | | P32 | MTIOC0C | | IRQ2 |
| 12 | | P17 | MTIOC0C/MTIOC3A/ MTIOC3B/POE8# | SCK1/MISOA/SDA0/RXD12/ RXDX12/SMISO12/SSCL12 | IRQ7 |
| 13 | | P16 | MTIOC3C/MTIOC3D | TXD1/SMOSI1/SSDA1/SCL0/ MOSIA/USB0_VBUSEN/ USB0_OVRCURB/USB0_VBUS | IRQ6/ADTRG0# |
| 14 | | P15 | MTIOC0B/MTCLKB | RXD1/SMISO1/SSCL1/RSPCKA | IRQ5/CLKOUT |
| 15 | UB# | P14 | MTIOC0A/MTIOC3A/ MTCLKA | CTS1#/RTS1#/SS1#/SSLA0/TXD12/ TXDX12/SIOX12/SMOSI12/ SSDA12/USB0_OVRCURA | IRQ4 |
| 16 | VCC_USB | | | | |
| 17 | | | | USB0_DM | |
| 18 | | | | USB0_DP | |
| 19 | VSS_USB | | | | |
| 20 | | PC4 | MTIOC3D/MTCLKC/POE0# | SCK5/SSLA0/USB0_VBUS*1/ USB0_VBUSEN | IRQ2/CLKOUT |
| 21 | | PB3 | MTIOC0A/MTIOC3B/ MTIOC4A/POE3# | USB0_OVRCURA | |
| 22 | VCC | | | | |
| 23 | | PB0 | MTIOC0C/MTIC5W | SCL0/RSPCKA | IRQ2/ADTRG0# |
| 24 | VSS | | | | |
| 25 | | PA6 | MTIOC2A/MTIC5V/MTCLKB/ POE2# | CTS5#/RTS5#/SS5#/SDA0/MOSIA | IRQ3 |
| 26 | | PA4 | MTIOC2B/MTIC5U/MTCLKA | TXD5/SMOSI5/SSDA5/SSLA0 | IRQ5 |
| 27 | | PA3 | MTIOC0D/MTIOC1B/ MTCLKD/POE0# | RXD5/SMISO5/SSCL5/MISOA | IRQ6 |
| 28 | | PA1 | MTIOC0B/MTCLKC | SCK5/SSLA2 | |
| 29 | | PE4 | MTIOC1A/MTIOC3A/ MTIOC4D | MOSIA | IRQ4/AN012 |
| 30 | | PE3 | MTIOC0A/MTIOC1B/ MTIOC4B/POE8# | CTS12#/RTS12#/SS12#/RSPCKA | IRQ3/AN011 |
| 31 | | PE2 | MTIOC4A | RXD12/RXDX12/SMISO12/SSCL12 | IRQ7/AN010 |
| 32 | | PE1 | MTIOC4C | TXD12/TXDX12/SIOX12/SMOSI12/ SSDA12 | IRQ1/AN009 |
| 33 | | PE0 | MTIOC2A/POE3# | SCK12 | IRQ0/AN008 |
| 34 | | P46 | | | AN006 |
| 35 | | P42 | | | AN002 |
| 36 | | P41 | | | AN001 |
| 37 | VREFL0 | PJ7 | | | |
| 38 | VREFH0 | PJ6 | | | |
| 39 | AVSS0 | | | | |

**Table 1.8 List of Pins and Pin Functions (40-Pin HWQFN) (2/2)**

| Pin No. | Power Supply, Clock, System Control | I/O Port | Timers (MTU, POE, RTC) | Communication (SCIe, SCIf, RSPI, RIIC, USB) | Others |
|---|---|---|---|---|---|
| 40 | AVCC0 | | | | |

Note 1. Not 5 V tolerant.

**Table 1.9 List of Pins and Pin Functions (36-Pin WFLGA)**

| Pin No. | Power Supply, Clock, System Control | I/O Port | Timers (MTU, POE, RTC) | Communication (SCIe, SCIf, RSPI, RIIC, USB) | Others |
|---|---|---|---|---|---|
| A1 | AVSS0 | | | | |
| A2 | AVCC0 | | | | |
| A3 | VREFH0 | PJ6 | | | |
| A4 | | P42 | | | AN002 |
| A5 | | P41 | | | AN001 |
| A6 | | PE2 | MTIOC4A | RXD12/RXDX12/SMISO12/SSCL12 | IRQ7/AN010 |
| B1 | RES# | | | | |
| B2 | | P27 | MTIOC2B | SCK1/SCK12 | IRQ3/CMPA2/CACREF/ ADTRG0# |
| B3 | VREFL0 | PJ7 | | | |
| B4 | | PE0 | MTIOC2A/POE3# | SCK12 | IRQ0/AN008 |
| B5 | | PE1 | MTIOC4C | TXD12/TXDX12/SIOX12/SMOSI12/ SSDA12 | IRQ1/AN009 |
| B6 | | PA3 | MTIOC0D/MTCLKD/ MTIOC1B/POE0# | RXD5/SMISO5/SSCL5/MISOA | IRQ6 |
| C1 | XTAL | | | | |
| C2 | MD | | | | FINED |
| C3 | | PE3 | MTIOC0A/MTIOC1B/ MTIOC4B/POE8# | CTS12#/RTS12#/SS12#/RSPCKA | IRQ3/AN011 |
| C4 | | PE4 | MTIOC1A/MTIOC3A/ MTIOC4D | MOSIA | IRQ4/AN012 |
| C5 | | PA4 | MTIOC2B/MTIC5U/MTCLKA | TXD5/SMOSI5/SSDA5/SSLA0 | IRQ5 |
| C6 | VSS | | | | |
| D1 | EXTAL | | | | |
| D2 | UPSEL | P35 | | | NMI |
| D3 | UB# | P14 | MTIOC0A/MTIOC3A/ MTCLKA | CTS1#/RTS1#/SS1#/SSLA0/TXD12/ TXDX12/SIOX12/SMOSI12/ SSDA12/USB0_OVRCURA | IRQ4 |
| D4 | | PA6 | MTIC5V/MTCLKB/MTIOC2A/ POE2# | CTS5#/RTS5#/SS5#/SDA0/MOSIA | IRQ3 |
| D5 | | PB3 | MTIOC0A/MTIOC3B/ MTIOC4A/POE3# | USB0_OVRCURA | |
| D6 | | PB0 | MTIOC0C/MTIC5W | SCL0/RSPCKA | IRQ2/ADTRG0# |
| E1 | VCL | | | | |
| E2 | | P17 | MTIOC0C/MTIOC3A/ MTIOC3B/POE8# | SCK1/MISOA/SDA0/RXD12/ RXDX12/SMISO12/SSCL12 | IRQ7 |
| E3 | | P16 | MTIOC3C/MTIOC3D | TXD1/SMOSI1/SSDA1/SCL0/ MOSIA/USB0_VBUSEN/ USB0_OVRCURB/USB0_VBUS | IRQ6/ADTRG0# |
| E4 | | P15 | MTIOC0B/MTCLKB | RXD1/SMISO1/SSCL1/RSPCKA | IRQ5/CLKOUT |
| E5 | | PC4 | MTIOC3D/MTCLKC/POE0# | SCK5/SSLA0/USB0_VBUSEN/ USB0_VBUS*1 | IRQ2/CLKOUT |
| E6 | VCC | | | | |
| F1 | VSS | | | | |
| F2 | VCC | | | | |
| F3 | VCC_USB | | | | |
| F4 | | | | USB0_DM | |
| F5 | | | | USB0_DP | |
| F6 | VSS_USB | | | | |

Note 1. Not 5 V tolerant.

# 2. CPU

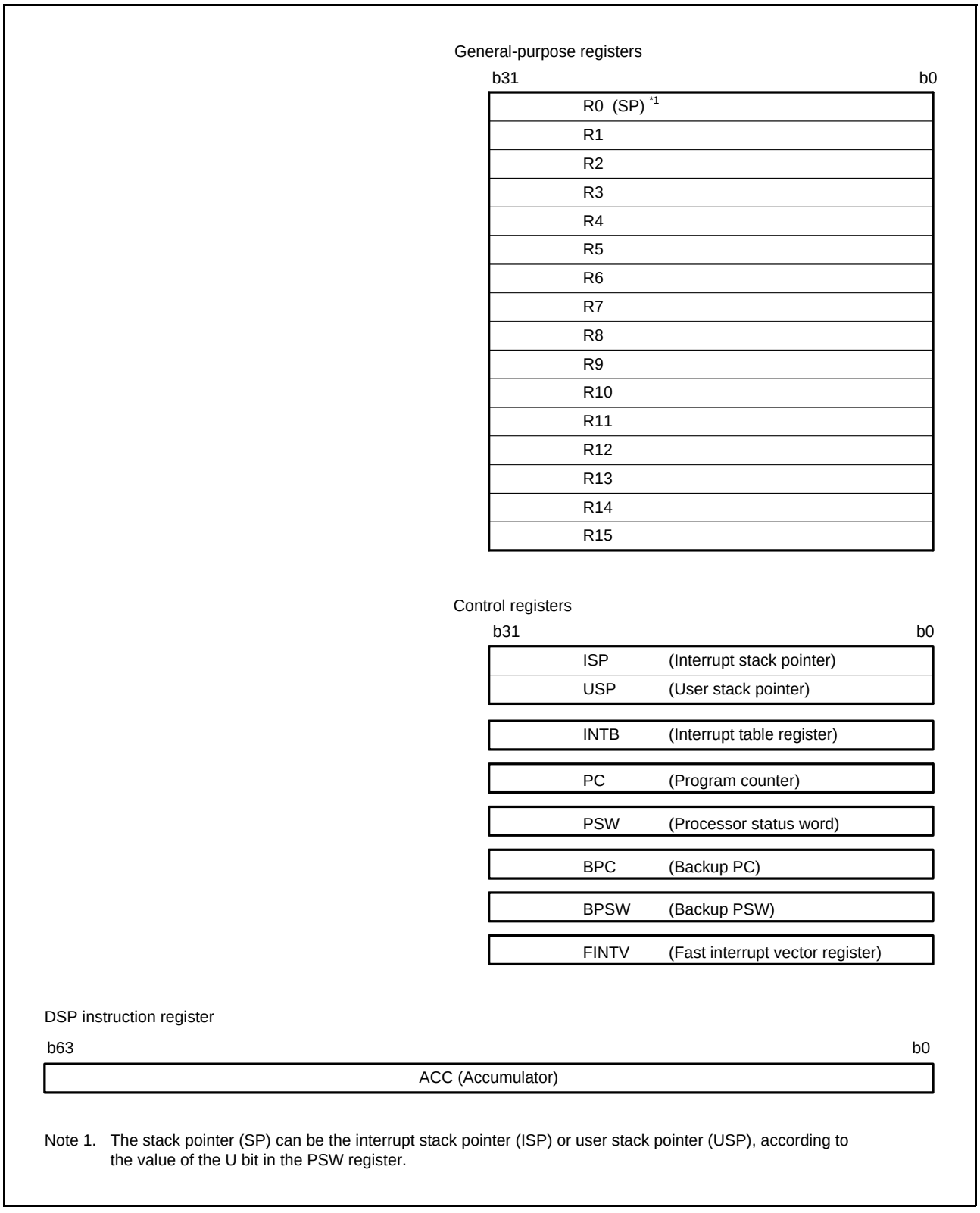
This MCU with the RX CPU as its core.
A variable-length instruction format has been adopted for the RX CPU. Allocating the more frequently used instructions to the shorter instruction lengths facilitates the development of efficient programs that take up less memory.
The CPU has 73 basic instructions and nine DSP instructions, for a total of 82 instructions. It has 10 addressing modes and caters to register-to-register operations, register-to-memory operations, immediate-to-register operations, immediate-to-memory operations, memory-to-memory transfer, and bitwise operations. In a single cycle, high-speed calculation is attained for not just register-to-register operations, but also for other types of combined instructions. The CPU includes an internal multiplier and an internal divider for high-speed multiplication and division.
The RX CPU has a five-stage pipeline for processing instructions. The stages are instruction fetching, decoding, execution, memory access, and write-back. In cases where pipeline processing is drawn-out by memory access, subsequent operations may in fact be executed earlier. By adopting an “out-of-order completion” of this kind, instruction execution is controlled to optimize the number of clock cycles.

## 2.1 Features

- Minimum instruction execution rate: One instruction per clock cycle
- Address space: 4-Gbyte linear
- Register set of the CPU
  General purpose: Sixteen 32-bit registers
  Control: Eight 32-bit registers
  Accumulator: One 64-bit register
- Basic instructions: 73 (arithmetic/logic instructions, data-transfer instructions, branch instructions, bit-manipulation instructions, string-manipulation instructions, and system-manipulation instructions)
  Relative branch instructions to suit branch distances
  Variable-length instruction format (lengths from 1 to 8 bytes)
  Short formats for frequently used instructions
- DSP instructions: 9
  Supports 16-bit × 16-bit multiplication and multiply-and-accumulate operations.
  Rounds the data in the accumulator.
- Addressing modes: 10
- Five-stage pipeline
  Adoption of “out-of-order completion”
- Processor modes
  A supervisor mode and a user mode are supported.
- Data arrangement
  Selectable as little endian or big endian

## 2.2 Register Set of the CPU

The RX CPU has 16 general-purpose registers, eight control registers, and one accumulator used for DSP instructions.




Note 1. The stack pointer (SP) can be the interrupt stack pointer (ISP) or user stack pointer (USP), according to the value of the U bit in the PSW register.

**Figure 2.1 Register Set of the CPU**

### 2.2.1 General-Purpose Registers (R0 to R15)

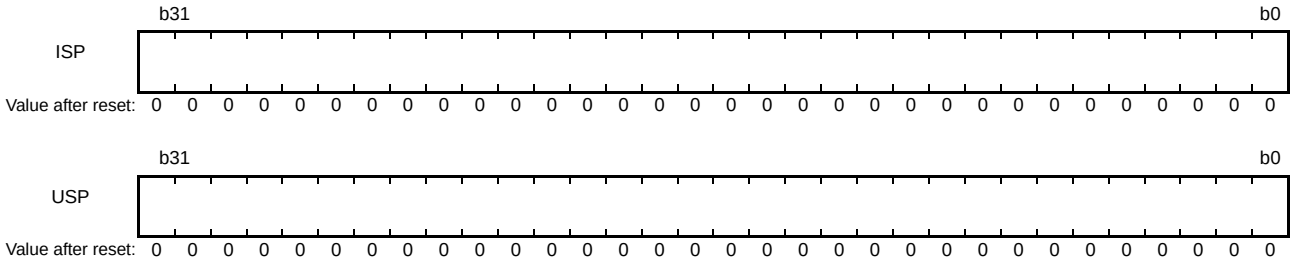
This CPU has 16 general-purpose registers (R0 to R15). R0 to R15 can be used as data registers or address registers. R0, a general-purpose register, also functions as the stack pointer (SP). The stack pointer is switched to operate as the interrupt stack pointer (ISP) or user stack pointer (USP) by the value of the stack pointer select bit (U) in the processor status word (PSW).

### 2.2.2 Control Registers

This CPU has the following eight control registers.

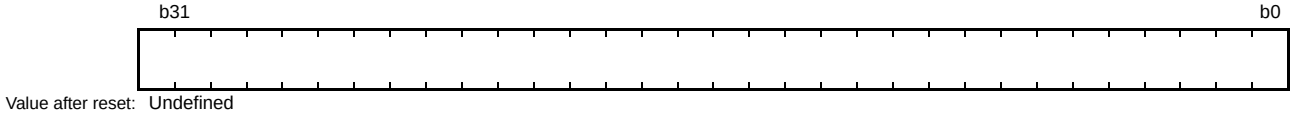
- Interrupt stack pointer (ISP)
- User stack pointer (USP)
- Interrupt table register (INTB)
- Program counter (PC)
- Processor status word (PSW)
- Backup PC (BPC)
- Backup PSW (BPSW)
- Fast interrupt vector register (FINTV)

### 2.2.2.1 Interrupt Stack Pointer (ISP)/User Stack Pointer (USP)



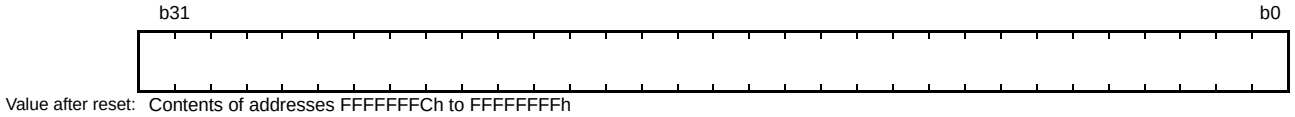

The stack pointer (SP) can be either of two types, the interrupt stack pointer (ISP) or the user stack pointer (USP). Whether the stack pointer operates as the ISP or USP depends on the value of the stack pointer select bit (U) in the processor status word (PSW).
Set the ISP or USP to a multiple of 4, as this reduces the numbers of cycles required to execute interrupt sequences and instructions entailing stack manipulation.

### 2.2.2.2 Interrupt Table Register (INTB)




The interrupt table register (INTB) specifies the address where the relocatable vector table starts.

### 2.2.2.3 Program Counter (PC)




The program counter (PC) indicates the address of the instruction being executed.

### 2.2.2.4 Processor Status Word (PSW)

| | b31 | b30 | b29 | b28 | b27 | b26 | b25 | b24 | b23 | b22 | b21 | b20 | b19 | b18 | b17 | b16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | IPL[3:0] | | | | — | — | — | PM | — | — | U | I |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | O | S | Z | C |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b0 | C | Carry Flag | 0: No carry has occurred.<br>1: A carry has occurred. | R/W |
| b1 | Z | Zero Flag | 0: Result is not 0.<br>1: Result is 0. | R/W |
| b2 | S | Sign Flag | 0: Result is a positive value or 0.<br>1: Result is a negative value. | R/W |
| b3 | O | Overflow Flag | 0: No overflow has occurred.<br>1: An overflow has occurred. | R/W |
| b15 to b4 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |
| b16 | I*1 | Interrupt Enable | 0: Interrupt disabled.<br>1: Interrupt enabled. | R/W |
| b17 | U*1 | Stack Pointer Select | 0: Interrupt stack pointer (ISP) is selected.<br>1: User stack pointer (USP) is selected. | R/W |
| b19, b18 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |
| b20 | PM*1,*2,*3 | Processor Mode Select | 0: Supervisor mode is selected.<br>1: User mode is selected. | R/W |
| b23 to b21 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |
| b27 to b24 | IPL[3:0]*1 | Processor Interrupt Priority Level | b27 b24<br>0 0 0 0: Priority level 0 (lowest)<br>0 0 0 1: Priority level 1<br>0 0 1 0: Priority level 2<br>0 0 1 1: Priority level 3<br>0 1 0 0: Priority level 4<br>0 1 0 1: Priority level 5<br>0 1 1 0: Priority level 6<br>0 1 1 1: Priority level 7<br>1 0 0 0: Priority level 8<br>1 0 0 1: Priority level 9<br>1 0 1 0: Priority level 10<br>1 0 1 1: Priority level 11<br>1 1 0 0: Priority level 12<br>1 1 0 1: Priority level 13<br>1 1 1 0: Priority level 14<br>1 1 1 1: Priority level 15 (highest) | R/W |
| b31 to b28 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |

Note 1. In user mode, writing to the IPL[3:0], PM, U, and I bits by an MVTC or a POPC instruction is ignored. Writing to the IPL[3:0] bits by an MVTIPL instruction generates a privileged instruction exception.
Note 2. In supervisor mode, writing to the PM bit by an MVTC or a POPC instruction is ignored, but writing to the other bits is possible.
Note 3. Switching from supervisor mode to user mode requires execution of an RTE instruction after having set the PSW.PM bit saved on the stack to 1 or executing an RTFI instruction after having set the BPSW.PM bit to 1.

The processor status word (PSW) indicates the results of instruction execution or the state of the CPU.

**C Flag (Carry Flag)**

This flag indicates whether a carry, borrow, or shift-out has occurred as the result of an operation.

**Z Flag (Zero Flag)**

This flag indicates that the result of an operation was 0.

**S Flag (Sign Flag)**

This flag indicates that the result of an operation was negative.

**O Flag (Overflow Flag)**

This flag indicates that an overflow occurred during an operation.

**I Bit (Interrupt Enable)**

This bit enables interrupt requests. When an exception is accepted, this bit becomes 0.

**U Bit (Stack Pointer Select)**

This bit specifies the stack pointer as either the ISP or USP. When an exception request is accepted, this bit becomes 0. When the processor mode is switched from supervisor mode to user mode, this bit becomes 1.

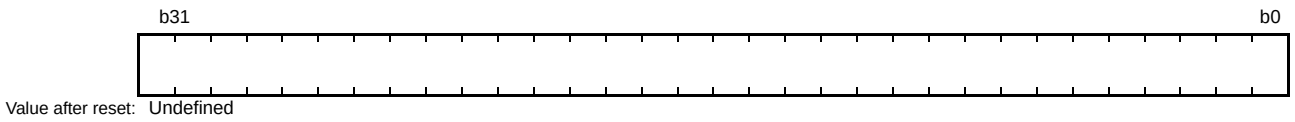
**PM Bit (Processor Mode Select)**

This bit specifies the processor mode. When an exception is accepted, this bit becomes 0.

**IPL[3:0] Bits (Processor Interrupt Priority Level)**

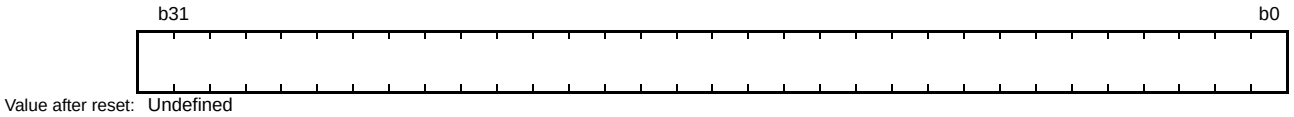
The IPL[3:0] bits specify the processor interrupt priority level as one of 16 levels from zero to 15, wherein priority level zero is the lowest and priority level 15 the highest. When the priority level of a requested interrupt is higher than the processor interrupt priority level, the interrupt is enabled. Setting the IPL[3:0] bits to level 15 (Fh) disables all interrupt requests. The IPL[3:0] bits are set to level 15 (Fh) when a non-maskable interrupt is generated. When interrupts are generated, the bits are set to the priority levels of accepted interrupts.

### 2.2.2.5 Backup PC (BPC)



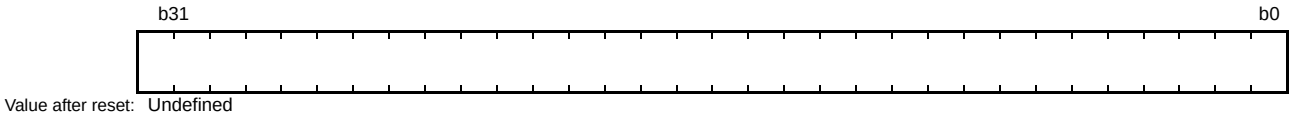

The backup PC (BPC) is provided to speed up response to interrupts.
After a fast interrupt has been generated, the contents of the program counter (PC) are saved in the BPC register.

### 2.2.2.6 Backup PSW (BPSW)




The backup PSW (BPSW) is provided to speed up response to interrupts.
After a fast interrupt has been generated, the contents of the processor status word (PSW) are saved in the BPSW. The allocation of bits in the BPSW corresponds to that in the PSW.

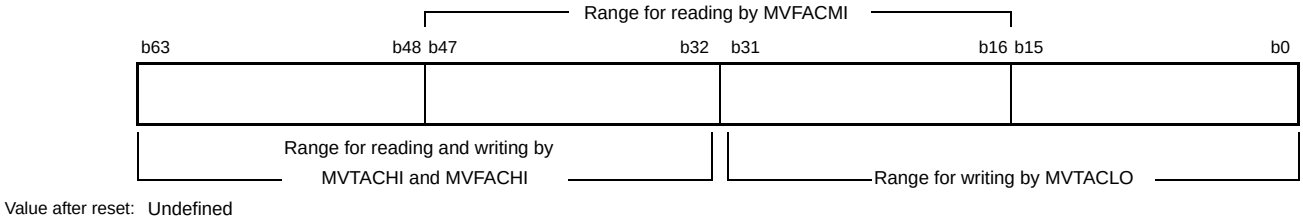
### 2.2.2.7 Fast Interrupt Vector Register (FINTV)




The fast interrupt vector register (FINTV) is provided to speed up response to interrupts.
The FINTV register specifies a branch destination address when a fast interrupt has been generated.

## 2.2.3 Register Associated with DSP Instructions

### 2.2.3.1 Accumulator (ACC)




The accumulator (ACC) is a 64-bit register used for DSP instructions. The accumulator is also used for the multiply and multiply-and-accumulate instructions; EMUL, EMULU, MUL, and RMPA, in which case the prior value in the accumulator is modified by execution of the instruction.
Use the MVTACHI and MVTACLO instructions for writing to the accumulator. The MVTACHI and MVTACLO instructions write data to the higher-order 32 bits (bits 63 to 32) and the lower-order 32 bits (bits 31 to 0), respectively.
Use the MVFACHI and MVFACMI instructions for reading data from the accumulator. The MVFACHI and MVFACMI instructions read data from the higher-order 32 bits (bits 63 to 32) and the middle 32 bits (bits 47 to 16), respectively.

## 2.3 Processor Mode

The RX CPU supports two processor modes, supervisor and user. These processor modes enable the realization of a hierarchical CPU resource protection.
Each processor mode imposes a level on rights of access to the CPU resources and the instructions that can be executed. Supervisor mode carries greater rights than those of user mode.
The initial state after a reset is supervisor mode.

### 2.3.1 Supervisor Mode

In supervisor mode, all CPU resources are accessible and all instructions are available. However, writing to the processor mode select bit (PM) in the processor status word (PSW) by executing an MVTC or a POPC instruction will be ignored. For details on how to write to the PM bit, refer to section 2.2.2.4, Processor Status Word (PSW).

### 2.3.2 User Mode

In user mode, write access to the CPU resources listed below is restricted. The restriction applies to any instruction capable of write access.

- Some bits (bits IPL[3:0], PM, U, and I) in the processor status word (PSW)
- Interrupt stack pointer (ISP)
- Interrupt table register (INTB)
- Backup PSW (BPSW)
- Backup PC (BPC)
- Fast interrupt vector register (FINTV)

### 2.3.3 Privileged Instruction

Privileged instructions can only be executed in supervisor mode. Executing a privileged instruction in user mode produces a privileged instruction exception. Privileged instructions include the RTFI, MVTIPL, RTE, and WAIT instructions.

### 2.3.4 Switching between Processor Modes

Manipulating the processor mode select bit (PM) in the processor status word (PSW) switches the processor mode. However, rewriting to the PM bit by executing an MVTC or a POPC instruction is prohibited. Switch the processor mode by following the procedures described below.

#### (1) Switching from user mode to supervisor mode

After an exception occurs, the PSW.PM bit is set to 0 and the CPU switches to supervisor mode. The hardware pre-processing is executed in supervisor mode. The state of the processor mode before the exception was generated is retained in the copy of PSW.PM bit is saved on the stack.

#### (2) Switching from supervisor mode to user mode

Executing an RTE instruction when the value of the copy of the PSW.PM bit that has been preserved on the stack is 1 or an RTFI instruction when the value of the copy of the PSW.PM bit that has been preserved in the backup PSW (BPSW) is 1 causes a transition to user mode. In the transition to user mode, the value of the stack pointer designation bit (the U bit in the PSW) becomes 1.

## 2.4 Data Types

The RX CPU can handle three types of data: integer, bit, and string.
For details, refer to RX Family User's Manual: Software.

## 2.5 Endian

For the RX CPU, instructions are little endian, but the data arrangement is selectable as little or big endian.

### 2.5.1 Switching the Endian

As arrangements of bytes, this MCU supports both big endian, where the higher-order byte (MSB) is at location 0, and little endian, where the lower-order byte (LSB) is at location 0.
For details on the endian setting, refer to section 3, Operating Modes.
Operations for access differ according to the endian setting and, depending on the instruction, whether 8-, 16- or 32-bit access has been selected. Operations for access in the various possible cases are described in Table 2.1 to Table 2.12.
In the tables,

LL indicates bits D7 to D0 of the general-purpose register,
LH indicates bits D15 to D8 of the general-purpose register,
HL indicates bits D23 to D16 of the general-purpose register, and
HH indicates bits D31 to D24 of the general-purpose register.

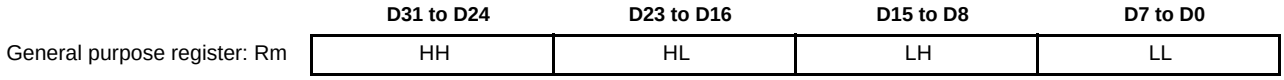



**Table 2.1 32-Bit Read Operations when Little Endian has been Selected**

| Operation / Address of src | Reading a 32-bit unit from address 0 | Reading a 32-bit unit from address 1 | Reading a 32-bit unit from address 2 | Reading a 32-bit unit from address 3 | Reading a 32-bit unit from address 4 |
|---|---|---|---|---|---|
| Address 0 | Transfer to LL | — | — | — | — |
| Address 1 | Transfer to LH | Transfer to LL | — | — | — |
| Address 2 | Transfer to HL | Transfer to LH | Transfer to LL | — | — |
| Address 3 | Transfer to HH | Transfer to HL | Transfer to LH | Transfer to LL | — |
| Address 4 | — | Transfer to HH | Transfer to HL | Transfer to LH | Transfer to LL |
| Address 5 | — | — | Transfer to HH | Transfer to HL | Transfer to LH |
| Address 6 | — | — | — | Transfer to HH | Transfer to HL |
| Address 7 | — | — | — | — | Transfer to HH |

**Table 2.2 32-Bit Read Operations when Big Endian has been Selected**

| Operation / Address of src | Reading a 32-bit unit from address 0 | Reading a 32-bit unit from address 1 | Reading a 32-bit unit from address 2 | Reading a 32-bit unit from address 3 | Reading a 32-bit unit from address 4 |
|---|---|---|---|---|---|
| Address 0 | Transfer to HH | — | — | — | — |
| Address 1 | Transfer to HL | Transfer to HH | — | — | — |
| Address 2 | Transfer to LH | Transfer to HL | Transfer to HH | — | — |
| Address 3 | Transfer to LL | Transfer to LH | Transfer to HL | Transfer to HH | — |
| Address 4 | — | Transfer to LL | Transfer to LH | Transfer to HL | Transfer to HH |
| Address 5 | — | — | Transfer to LL | Transfer to LH | Transfer to HL |
| Address 6 | — | — | — | Transfer to LL | Transfer to LH |
| Address 7 | — | — | — | — | Transfer to LL |

**Table 2.3 32-Bit Write Operations when Little Endian has been Selected**

| Operation / Address of dest | Writing a 32-bit unit to address 0 | Writing a 32-bit unit to address 1 | Writing a 32-bit unit to address 2 | Writing a 32-bit unit to address 3 | Writing a 32-bit unit to address 4 |
|---|---|---|---|---|---|
| Address 0 | Transfer from LL | — | — | — | — |
| Address 1 | Transfer from LH | Transfer from LL | — | — | — |
| Address 2 | Transfer from HL | Transfer from LH | Transfer from LL | — | — |
| Address 3 | Transfer from HH | Transfer from HL | Transfer from LH | Transfer from LL | — |
| Address 4 | — | Transfer from HH | Transfer from HL | Transfer from LH | Transfer from LL |
| Address 5 | — | — | Transfer from HH | Transfer from HL | Transfer from LH |
| Address 6 | — | — | — | Transfer from HH | Transfer from HL |
| Address 7 | — | — | — | — | Transfer from HH |

**Table 2.4 32-Bit Write Operations when Big Endian has been Selected**

| Operation / Address of dest | Writing a 32-bit unit to address 0 | Writing a 32-bit unit to address 1 | Writing a 32-bit unit to address 2 | Writing a 32-bit unit to address 3 | Writing a 32-bit unit to address 4 |
|---|---|---|---|---|---|
| Address 0 | Transfer from HH | — | — | — | — |
| Address 1 | Transfer from HL | Transfer from HH | — | — | — |
| Address 2 | Transfer from LH | Transfer from HL | Transfer from HH | — | — |
| Address 3 | Transfer from LL | Transfer from LH | Transfer from HL | Transfer from HH | — |
| Address 4 | — | Transfer from LL | Transfer from LH | Transfer from HL | Transfer from HH |
| Address 5 | — | — | Transfer from LL | Transfer from LH | Transfer from HL |
| Address 6 | — | — | — | Transfer from LL | Transfer from LH |
| Address 7 | — | — | — | — | Transfer from LL |

**Table 2.5 16-Bit Read Operations when Little Endian has been Selected**

| Operation / Address of src | Reading a 16-bit unit from address 0 | Reading a 16-bit unit from address 1 | Reading a 16-bit unit from address 2 | Reading a 16-bit unit from address 3 | Reading a 16-bit unit from address 4 | Reading a 16-bit unit from address 5 | Reading a 16-bit unit from address 6 |
|---|---|---|---|---|---|---|---|
| Address 0 | Transfer to LL | — | — | — | — | — | — |
| Address 1 | Transfer to LH | Transfer to LL | — | — | — | — | — |
| Address 2 | — | Transfer to LH | Transfer to LL | — | — | — | — |
| Address 3 | — | — | Transfer to LH | Transfer to LL | — | — | — |
| Address 4 | — | — | — | Transfer to LH | Transfer to LL | — | — |
| Address 5 | — | — | — | — | Transfer to LH | Transfer to LL | — |
| Address 6 | — | — | — | — | — | Transfer to LH | Transfer to LL |
| Address 7 | — | — | — | — | — | — | Transfer to LH |

**Table 2.6 16-Bit Read Operations when Big Endian has been Selected**

| Operation / Address of src | Reading a 16-bit unit from address 0 | Reading a 16-bit unit from address 1 | Reading a 16-bit unit from address 2 | Reading a 16-bit unit from address 3 | Reading a 16-bit unit from address 4 | Reading a 16-bit unit from address 5 | Reading a 16-bit unit from address 6 |
|---|---|---|---|---|---|---|---|
| Address 0 | Transfer to LH | — | — | — | — | — | — |
| Address 1 | Transfer to LL | Transfer to LH | — | — | — | — | — |
| Address 2 | — | Transfer to LL | Transfer to LH | — | — | — | — |
| Address 3 | — | — | Transfer to LL | Transfer to LH | — | — | — |
| Address 4 | — | — | — | Transfer to LL | Transfer to LH | — | — |
| Address 5 | — | — | — | — | Transfer to LL | Transfer to LH | — |
| Address 6 | — | — | — | — | — | Transfer to LL | Transfer to LH |
| Address 7 | — | — | — | — | — | — | Transfer to LL |

**Table 2.7 16-Bit Write Operations when Little Endian has been Selected**

| Operation / Address of dest | Writing a 16-bit unit to address 0 | Writing a 16-bit unit to address 1 | Writing a 16-bit unit to address 2 | Writing a 16-bit unit to address 3 | Writing a 16-bit unit to address 4 | Writing a 16-bit unit to address 5 | Writing a 16-bit unit to address 6 |
|---|---|---|---|---|---|---|---|
| Address 0 | Transfer from LL | — | — | — | — | — | — |
| Address 1 | Transfer from LH | Transfer from LL | — | — | — | — | — |
| Address 2 | — | Transfer from LH | Transfer from LL | — | — | — | — |
| Address 3 | — | — | Transfer from LH | Transfer from LL | — | — | — |
| Address 4 | — | — | — | Transfer from LH | Transfer from LL | — | — |
| Address 5 | — | — | — | — | Transfer from LH | Transfer from LL | — |
| Address 6 | — | — | — | — | — | Transfer from LH | Transfer from LL |
| Address 7 | — | — | — | — | — | — | Transfer from LH |

**Table 2.8 16-Bit Write Operations when Big Endian has been Selected**

| Operation / Address of dest | Writing a 16-bit unit to address 0 | Writing a 16-bit unit to address 1 | Writing a 16-bit unit to address 2 | Writing a 16-bit unit to address 3 | Writing a 16-bit unit to address 4 | Writing a 16-bit unit to address 5 | Writing a 16-bit unit to address 6 |
|---|---|---|---|---|---|---|---|
| Address 0 | Transfer from LL | — | — | — | — | — | — |
| Address 1 | Transfer from LH | Transfer from LL | — | — | — | — | — |
| Address 2 | — | Transfer from LH | Transfer from LL | — | — | — | — |
| Address 3 | — | — | Transfer from LH | Transfer from LL | — | — | — |
| Address 4 | — | — | — | Transfer from LH | Transfer from LL | — | — |
| Address 5 | — | — | — | — | Transfer from LH | Transfer from LL | — |
| Address 6 | — | — | — | — | — | Transfer from LH | Transfer from LL |
| Address 7 | — | — | — | — | — | — | Transfer from LH |

**Table 2.9 8-Bit Read Operations when Little Endian has been Selected**

| Operation / Address of src | Reading an 8-bit unit from address 0 | Reading an 8-bit unit from address 1 | Reading an 8-bit unit from address 2 | Reading an 8-bit unit from address 3 |
|---|---|---|---|---|
| Address 0 | Transfer to LL | — | — | — |
| Address 1 | — | Transfer to LL | — | — |
| Address 2 | — | — | Transfer to LL | — |
| Address 3 | — | — | — | Transfer to LL |

**Table 2.10 8-Bit Read Operations when Big Endian has been Selected**

| Operation / Address of src | Reading an 8-bit unit from address 0 | Reading an 8-bit unit from address 1 | Reading an 8-bit unit from address 2 | Reading an 8-bit unit from address 3 |
|---|---|---|---|---|
| Address 0 | Transfer to LL | — | — | — |
| Address 1 | — | Transfer to LL | — | — |
| Address 2 | — | — | Transfer to LL | — |
| Address 3 | — | — | — | Transfer to LL |

**Table 2.11 8-Bit Write Operations when Little Endian has been Selected**

| Operation / Address of dest | Writing an 8-bit unit to address 0 | Writing an 8-bit unit to address 1 | Writing an 8-bit unit to address 2 | Writing an 8-bit unit to address 3 |
|---|---|---|---|---|
| Address 0 | Transfer from LL | — | — | — |
| Address 1 | — | Transfer from LL | — | — |
| Address 2 | — | — | Transfer from LL | — |
| Address 3 | — | — | — | Transfer from LL |

**Table 2.12 8-Bit Write Operations when Big Endian has been Selected**

| Operation / Address of dest | Writing an 8-bit unit to address 0 | Writing an 8-bit unit to address 1 | Writing an 8-bit unit to address 2 | Writing an 8-bit unit to address 3 |
|---|---|---|---|---|
| Address 0 | Transfer from LL | — | — | — |
| Address 1 | — | Transfer from LL | — | — |
| Address 2 | — | — | Transfer from LL | — |
| Address 3 | — | — | — | Transfer from LL |

### 2.5.2 Access to I/O Registers

The addresses of I/O registers are fixed, and this is regardless of whether the setting is for little endian or big endian. Accordingly, changes to the endian do not affect access to I/O registers. For the arrangements of I/O registers, refer to the descriptions of registers in the relevant sections.

### 2.5.3 Notes on Access to I/O Registers

Ensure that access to I/O registers is in accord with the following rules.

- With I/O registers for which a bus width of 8 bits is indicated, use instructions having operands of the same width (8 bits). That is, access these registers by using instructions with .B as the size specifier (.size), or with .B or .UB as the size-extension specifier (.memex).
- With I/O registers for which a bus width of 16 bits is indicated, use instructions having operands of the same width (16 bits). That is, access these registers by using instructions with .W as the size specifier (.size), or with .W or .UW as the size-extension specifier (.memex).
- With I/O registers for which a bus width of 32 bits is indicated, use instructions having operands of the same width (32 bits). That is, access these registers by using instructions with .L as the size specifier (.size), or with .L size-extension specifier (.memex).

### 2.5.4 Data Arrangement

#### 2.5.4.1 Data Arrangement in Registers

Figure 2.2 shows the relation between the sizes of registers and bit numbers.

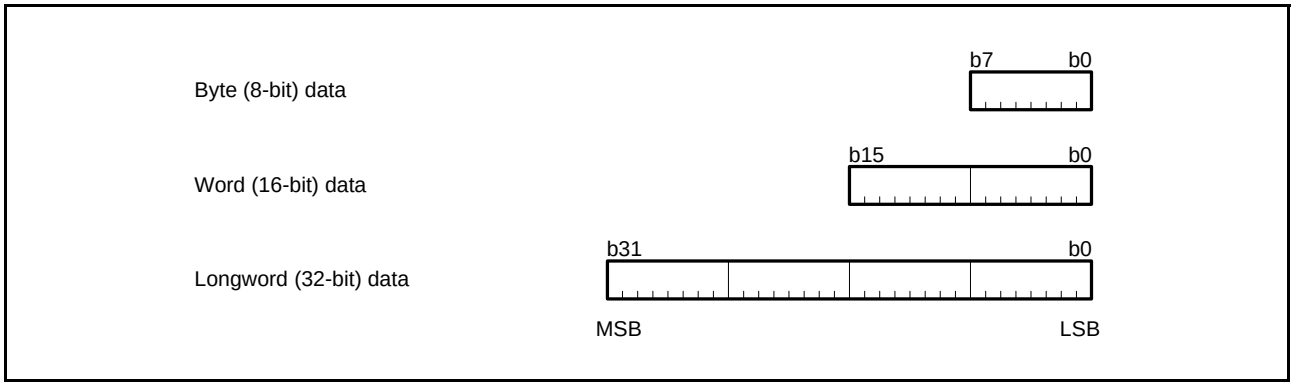



**Figure 2.2 Data Arrangement in Registers**

### 2.5.4.2 Data Arrangement in Memory

Data in memory have three sizes: byte (8-bit), word (16-bit), and longword (32-bit). The data arrangement is selectable as little endian or big endian. Figure 2.3 shows the arrangement of data in memory.

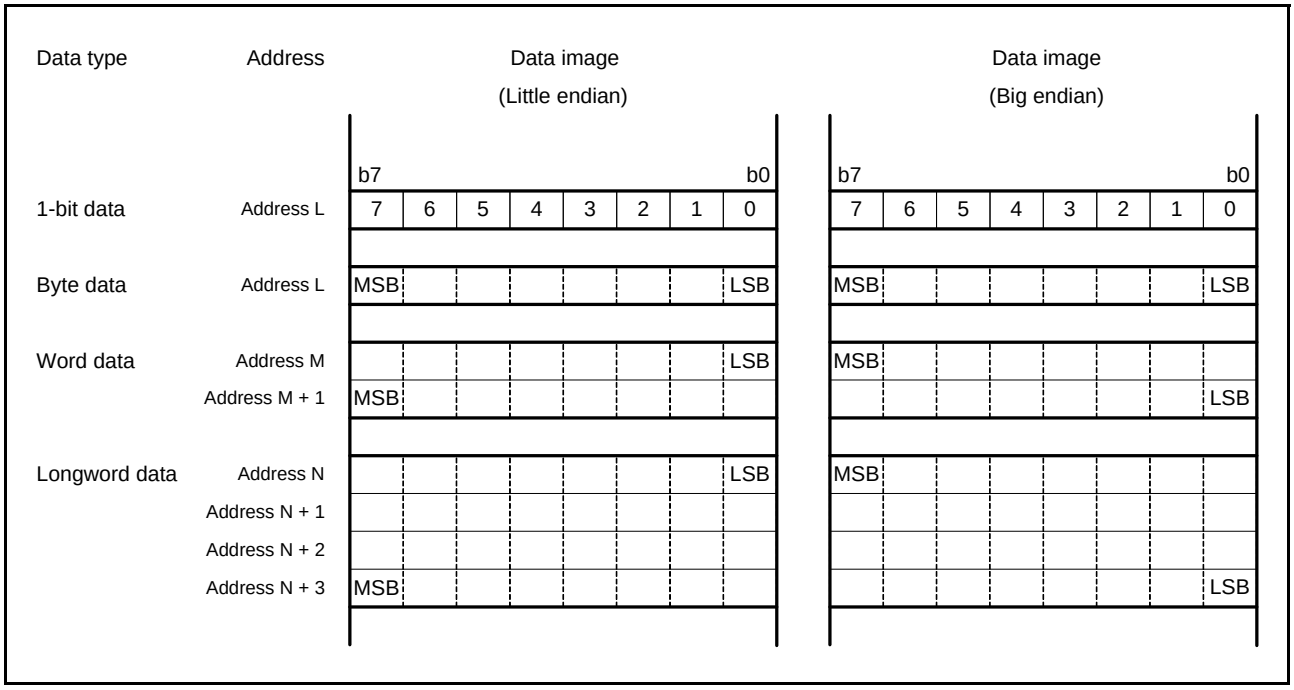



**Figure 2.3 Data Arrangement in Memory**

### 2.5.5 Notes on the Allocation of Instruction Codes

The allocation of instruction codes to an external space where the endian differs from that of the chip is prohibited. If the instruction codes are allocated to the external space, they must be allocated to areas where the endian setting is the same as that for the chip.

## 2.6 Vector Table

There are two types of vector table: fixed and relocatable. Each vector in the vector table consists of 4 bytes and specifies the address where the corresponding exception handling routine starts.

### 2.6.1 Fixed Vector Table

The fixed vector table is allocated to a fixed address range. The individual vectors for the privileged instruction exception, undefined instruction exception, non-maskable interrupt, and reset are allocated to addresses in the range from FFFFFF80h to FFFFFFFFh. Figure 2.4 shows the fixed vector table.

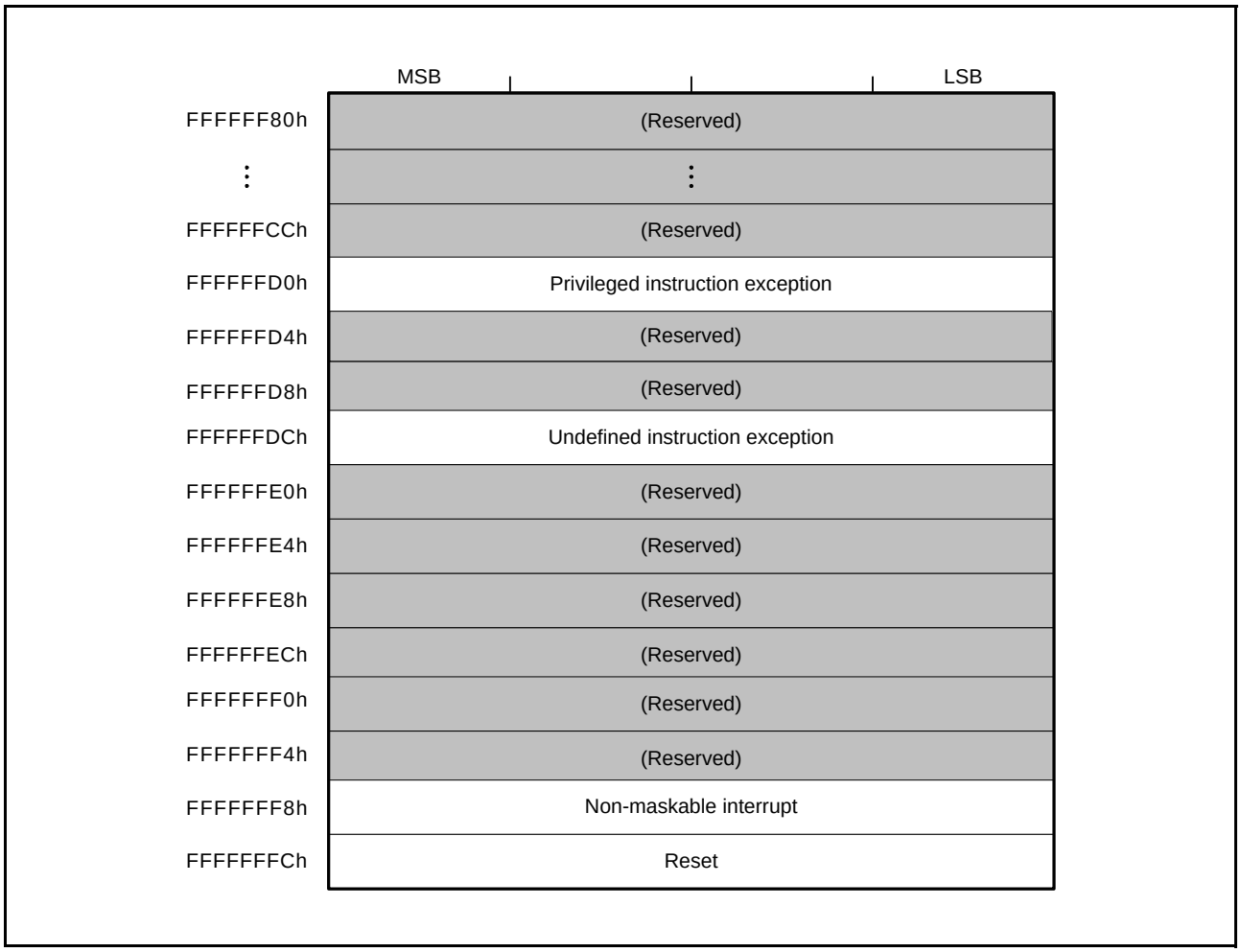



**Figure 2.4 Fixed Vector Table**

### 2.6.2 Relocatable Vector Table

The address where the relocatable vector table is placed can be adjusted. The table is a 1,024-byte region that contains all vectors for unconditional traps and interrupts and starts at the address (IntBase) specified in the interrupt table register (INTB). Figure 2.5 shows the relocatable vector table.

Each vector in the relocatable vector table has a vector number from 0 to 255. Each of the INT instructions, which act as the sources of unconditional traps, is allocated to the vector that has the same number as is specified as the operand of the instruction itself (from 0 to 255). The BRK instruction is allocated to the vector with number 0. Furthermore, vector numbers (from 0 to 255) are allocated to interrupt requests in a fixed way for each product. For more on interrupt vector numbers, refer to section 14.3.1, Interrupt Vector Table.

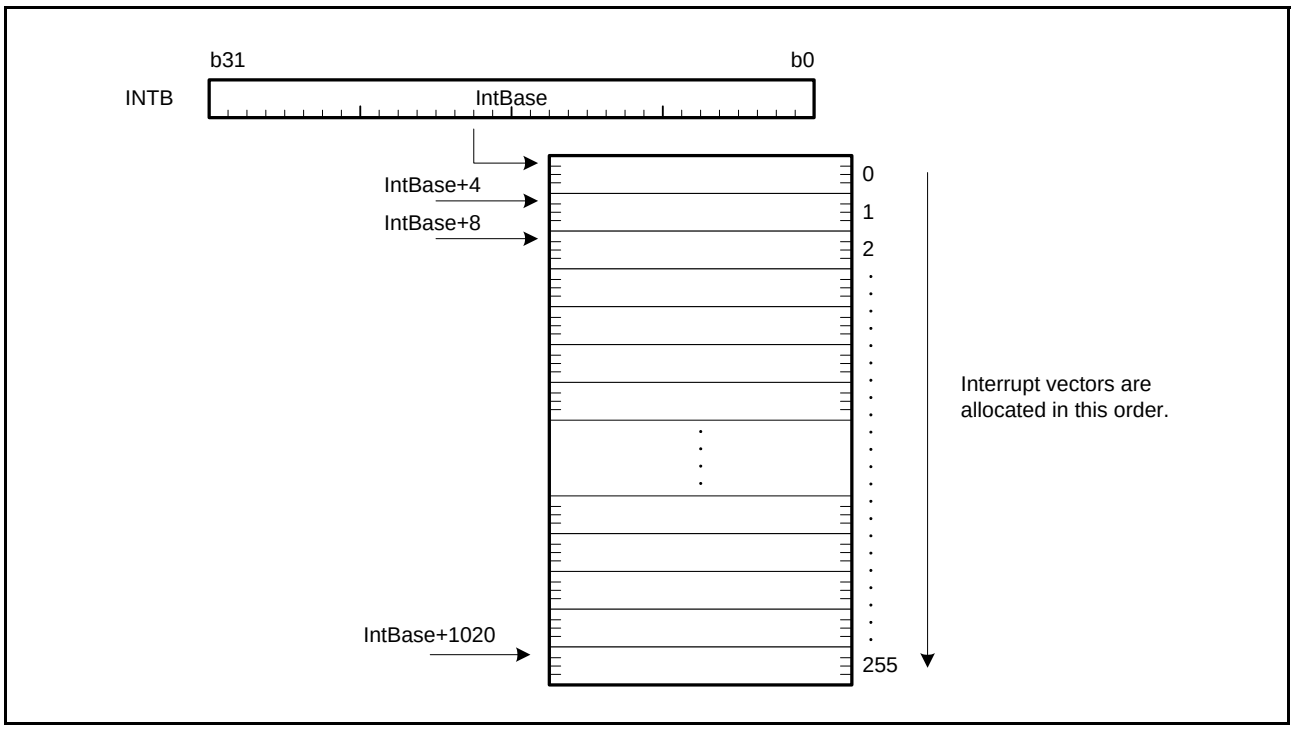



**Figure 2.5 Relocatable Vector Table**

## 2.7 Operation of Instructions

### 2.7.1 Data Prefetching by the RMPA Instruction and the String-Manipulation Instructions

The RMPA instruction and the string-manipulation instructions except the SSTR instruction (that is, SCMPU, SMOVB, SMOVF, SMOVU, SUNTIL, and SWHILE instructions) may prefetch data from the memory to speed up the read processing. Data is prefetched from the prefetching start position with 3 bytes as the upper limit. The prefetching start positions of each operation are shown below.

- RMPA instruction: The multiplicand address specified by R1, and the multiplier address specified by R2
- SCMPU instruction: The source address specified by R1 for comparison, and the destination address specified by R2 for comparison
- SUNTIL and SWHILE instructions: The destination address specified by R1 for comparison
- SMOVB, SMOVF, and SMOVU instructions: The source address specified by R2 for transfer

## 2.8 Pipeline

### 2.8.1 Overview

The RX CPU has five-stage pipeline structure. The RX CPU instruction is converted into one or more micro-operations, which are then executed in pipeline processing. In the pipeline stage, the IF stage is executed in the unit of instructions, while the D and subsequent stages are executed in the unit of micro-operations.
The operation of pipeline and respective stages is described below.

#### (1) IF stage (instruction fetch stage)

In the IF stage, the CPU fetches instructions from the memory. As the RX CPU has four 4-byte instruction queues, it fetches instructions until the instruction queue is full, regardless of the completion of decoding in the D (decoding) stage.

#### (2) D stage (decoding stage)

The CPU decodes instructions (DEC) in the D stage and converts them into micro-operations. The CPU reads the register information (RF) in this stage and executes a bypass process (BYP) if the result of the preceding instruction will be used in a subsequent instruction. The write of operation result to the register (RW) can be executed with the register reference by using the bypass process.

#### (3) E stage (execution stage)

Operations and address calculations (OP) are processed in the E stage.

#### (4) M stage (memory access stage)

Operand memory accesses (OA1, OA2) are processed in the M stage. This stage is used only when the memory is accessed, and is divided into two sub-stages, M1 and M2. The RX CPU enables respective memory accesses for M1 and M2.

- M1 stage (memory-access stage 1)
  Operand memory access (OA1) is processed.
  Store operation: The pipeline processing ends when a write request is received via the bus.
  Load operation: The operation proceeds to the M2 stage when a read request is received via the bus. If a request and load data are received at the same timing (no-wait memory access), the operation proceeds to the WB stage.
- M2 stage (memory-access stage 2)
  Operand memory access (OA2) is processed. The CPU waits for the load data in the M2 stage. When the load data is received, the operation proceeds to the WB stage.

### (5) WB stage (write-back stage)

The operation result and the data read from memory are written to the register (RW) in the WB stage. The data read from memory and the other type of data, such as the operation result, can be written to the register in the same clock cycles.

Figure 2.6 shows the pipeline configuration and its operation.

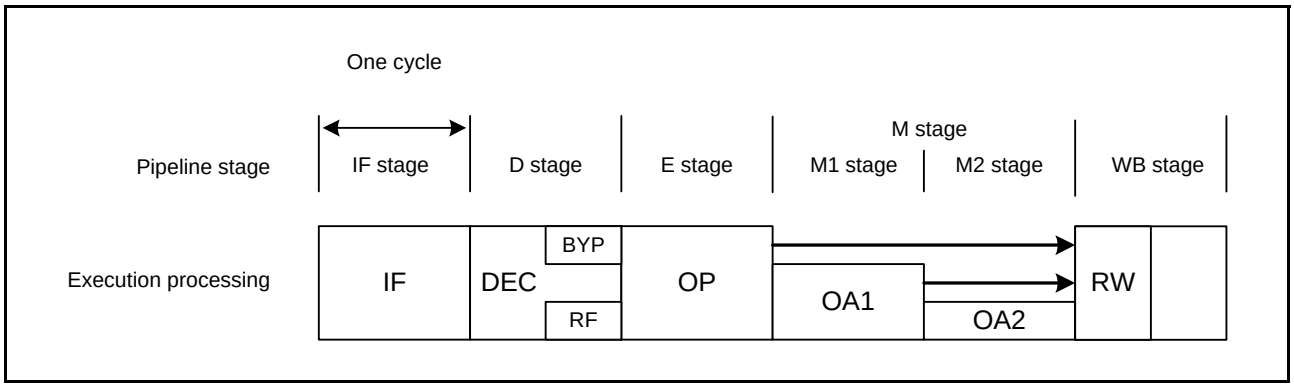



**Figure 2.6 Pipeline Configuration and its Operation**

### 2.8.2 Instructions and Pipeline Processing

The operands in the table below indicate the following meaning.

#IMM: Immediate
flag: bit, flag
Rs, Rs2, Rd, Rd2, Ri, Rb: General-purpose register
CR: Control register
dsp: displacement
pcdsp: displacement

#### 2.8.2.1 Instructions Converted into Single Micro-Operation and Pipeline Processing

The table below lists the instructions that are converted into a single micro-operation. The number of cycles in the table indicates the number of cycles during no-wait memory access.

**Table 2.13 Instructions that are Converted into a Single Micro-Operation**

| Instruction | Mnemonic (indicates the common operation when the size is omitted) | Reference Figure | Number of Cycles |
|---|---|---|---|
| Arithmetic/logic instructions (register-register, immediate-register) Except EMUL, EMULU, RMPA, DIV, DIVU and SATR | • {ABS, NEG, NOT} “Rd”/“Rs, Rd”<br>• {ADC, MAX, MIN, ROTL, ROTR, XOR} “#IMM, Rd”/“Rs, Rd”<br>• ADD “#IMM, Rd”/“Rs, Rd”/“#IMM, Rs, Rd”/“Rs, Rs2, Rd”<br>• {AND, MUL, OR, SUB} “#IMM, Rd”/“Rs, Rd”/“Rs, Rs2, Rd”<br>• {CMP, TST} “#IMM, Rs”/“Rs, Rs2”<br>• NOP<br>• {ROLC, RORC, SAT} “Rd”<br>• SBB “Rs, Rd”<br>• {SHAR, SHLL, SHLR} “#IMM, Rd”/“Rs, Rd”/“#IMM, Rs, Rd” | Figure 2.7 | 1 |
| Arithmetic/logic instructions (division) | • DIV “#IMM, Rd”/“Rs, Rd” | Figure 2.7 | 3 to 20*1 |
| | • DIVU “#IMM, Rd”/“Rs, Rd” | Figure 2.7 | 2 to 18*1 |
| Data transfer instructions (register-register, immediate-register) | • MOV “#IMM, Rd”/“Rs, Rd”<br>• {MOVU, REVL, REVW} “Rs, Rd”<br>• SC*Cnd* “Rd”<br>• {STNZ, STZ} “#IMM, Rd” | Figure 2.7 | 1 |
| Transfer instructions (load operation) | • {MOV, MOVU} “[Rs], Rd”/“dsp[Rs], Rd”/“[Rs+], Rd”/“[-Rs], Rd”/“[Ri, Rb], Rd”<br>• POP “Rd” | Figure 2.8 | Throughput: 1<br>Latency: 2*2 |
| Transfer instructions (store operation) | • MOV “Rs, [Rd]”/“Rs, dsp[Rd]”/“Rs, [Rd+]”/“Rs, [-Rd]”/“Rs, [Ri, Rb]”/“#IMM, dsp[Rd]”/“#IMM, [Rd]”<br>• PUSH “Rs”<br>• PUSHC “CR”<br>• SCCnd “[Rd]”/“dsp[Rd]” | Figure 2.9 | 1 |
| Bit manipulation instructions (register) | • {BCLR, BNOT, BSET} “#IMM, Rd”/“Rs, Rd”<br>• BM*Cnd* “#IMM, Rd”<br>• BTST “#IMM, Rs”/“Rs, Rs2” | Figure 2.7 | 1 |
| Branch instructions | • B*Cnd* “pcdsp”<br>• {BRA, BSR} “pcdsp”/“Rs”<br>• {JMP, JSR} “Rs” | Figure 2.17 | Branch taken: 3<br>Branch not taken: 1 |
| System manipulation instructions | • {CLRPSW, SETPSW} “flag”<br>• MVTC “#IMM, CR”/“Rs, CR”<br>• MVFC “CR, Rd”<br>• MVTIPL“#IMM” | — | 1 |
| DSP instructions | • {MACHI, MACLO, MULHI, MULLO} “Rs, Rs2”<br>• {MVFACHI, MVFACMI} “Rd”<br>• {MVTACHI, MVTACLO} “Rs”<br>• RACW“#IMM” | Figure 2.7 | 1 |

Note 1. The number of cycles for the dividing instruction varies according to the divisor and dividend.
Note 2. For the number of cycles for throughput and latency, refer to section 2.8.3, Calculation of the Instruction Processing Time.

Figure 2.7 to Figure 2.9 show the operation of instructions that are converted into a basic single micro-operation.

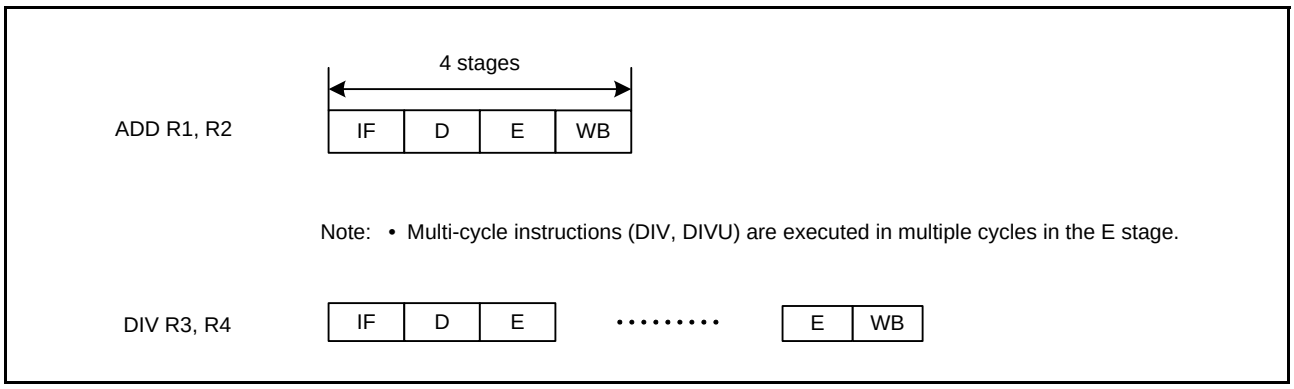



**Figure 2.7 Operation for Register-Register, Immediate-Register**

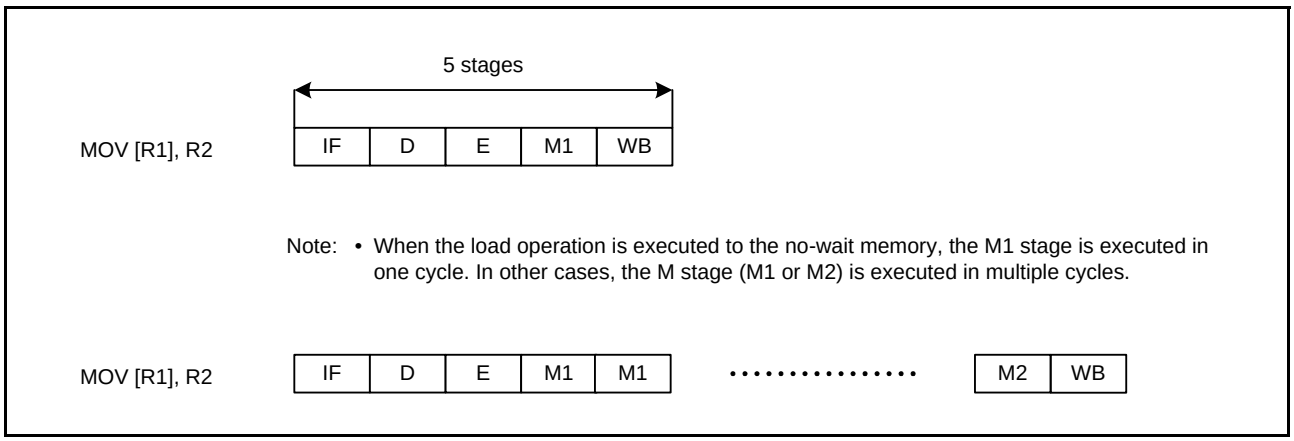



**Figure 2.8 Load Operation**

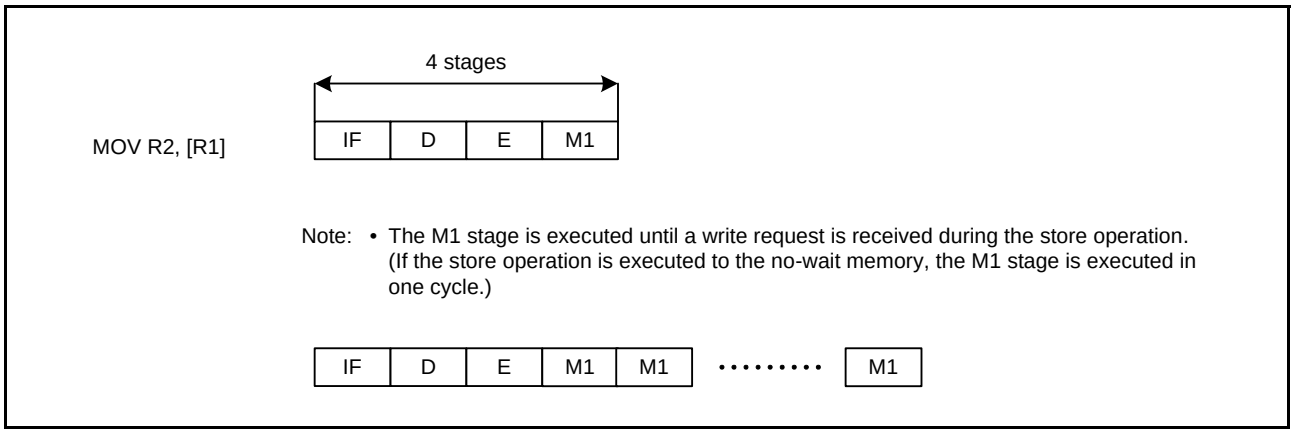



**Figure 2.9 Store Operation**

### 2.8.2.2 Instructions Converted into Multiple Micro-Operations and Pipeline Processing

The table below lists the instructions that are converted into multiple micro-operations. The number of cycles in the table indicates the number of cycles during no-wait memory access.

**Table 2.14 Instructions that are Converted into Multiple Micro-Operations (1/2)**

| Instruction | Mnemonic (indicates the common operation when the size is omitted) | Reference Figure | Number of Cycles |
|---|---|---|---|
| Arithmetic/logic instructions (memory source operand) | • {ADC, ADD, AND, MAX, MIN, MUL, OR, SBB, SUB, XOR} “[Rs], Rd”/“dsp[Rs], Rd”<br>• {CMP, TST} “[Rs], Rs2”/“dsp[Rs], Rs2” | Figure 2.10 | 3 |
| Arithmetic/logic instructions (division) | • DIV “[Rs], Rd / dsp[Rs], Rd” | — | 5 to 22 |
| | • DIVU“[Rs], Rd / dsp[Rs], Rd” | — | 4 to 20 |
| Arithmetic/logic instruction (multiplier: 32 × 32 → 64 bits) (register-register, register-immediate) | • {EMUL, EMULU} “#IMM, Rd”/“Rs, Rd” | Figure 2.12 | 2 |
| Arithmetic/logic instruction (multiplier: 32 × 32 → 64 bits) (memory source operand) | • {EMUL, EMULU} “[Rs], Rd”/“dsp[Rs], Rd” | — | 4 |
| Arithmetic/logic instructions (multiply-and-accumulate operation) | • RMPA.B | — | 6+7×floor(n/4)+4×(n%4)<br>n: Number of processing bytes*1 |
| | • RMPA.W | — | 6+5×floor(n/2)+4×(n%2)<br>n: Number of processing words*1 |
| | • RMPA.L | — | 6+4n<br>n: Number of processing longwords*1 |
| Arithmetic/logic instruction (64-bit signed saturation processing for the RMPA instruction) | • SATR | — | 3 |
| Data transfer instructions (memory-memory transfer) | • MOV “[Rs], [Rd]”/“dsp[Rs], [Rd]”/“[Rs], dsp[Rd]”/“dsp[Rs], dsp[Rd]”<br>• PUSH “[Rs]”/“dsp[Rs]” | Figure 2.11 | 3 |
| Bit manipulation instructions (memory source operand) | • {BCLR, BNOT, BSET} “#IMM, [Rd]”/“#IMM, dsp[Rd]”/“Rs, [Rd]”/“Rs, dsp[Rd]”<br>• BM*Cnd* “#IMM, [Rd]”/“#IMM, dsp[Rd]”<br>• BTST “#IMM, [Rs]”/“#IMM, dsp[Rs]”/“Rs, [Rs2]”/“Rs, dsp[Rs2] | Figure 2.11 | 3 |
| Transfer instruction (load operation) | • POPC “CR” | — | Throughput: 3<br>Latency: 4*2 |
| Transfer instruction (save operation of multiple registers) | • PUSHM “Rs-Rs2” | — | n<br>n: Number of registers*3 |
| Transfer instruction (restore operation of multiple registers) | • POPM “Rs-Rs2” | — | Throughput: n<br>Latency: n+1<br>n: Number of registers*2,*4 |
| Transfer instruction (register-register) | • XCHG “Rs, Rd” | Figure 2.13 | 2 |
| Transfer instruction (memory-register) | • XCHG “[Rs], Rd”/“dsp[Rs], Rd” | Figure 2.14 | 2 |
| Branch instructions | • RTS | — | 5 |
| | • RTSD “#IMM” | — | 5 |
| | • RTSD “#IMM, Rd-Rd2” | — | Throughput: n<5?5:1+n<br>Latency: n<4?5:2+n<br>n: Number of registers*2 |

**Table 2.14 Instructions that are Converted into Multiple Micro-Operations (2/2)**

| Instruction | Mnemonic (indicates the common operation when the size is omitted) | Reference Figure | Number of Cycles |
|---|---|---|---|
| String manipulation instructions*5 | • SCMPU | — | 2+4×floor(n/4)+4×(n%4)<br>n: Number of comparison bytes*1 |
| | • SMOVB | — | n>3?<br>6+3×floor(n/4)+3×(n%4):<br>2+3n<br>n: Number of transfer bytes*1 |
| | • SMOVF, SMOVU | — | 2+3×floor(n/4)+3×(n%4)<br>n: Number of transfer bytes*1 |
| | • SSTR.B | — | 2+floor(n/4)+n%4<br>n: Number of transfer bytes*1 |
| | • SSTR.W | — | 2+floor(n/2)+n%2<br>n: Number of transfer words*1 |
| | • SSTR.L | — | 2+n<br>n: Number of transfer longwords |
| | • SUNTIL.B, SWHILE.B | — | 3+3×floor(n/4)+3×(n%4)<br>n: Number of comparison bytes*1 |
| | • SUNTIL.W, SWHILE.W | — | 3+3×floor(n/2)+3×(n%2)<br>n: Number of comparison words*1 |
| | • SUNTIL.L, SWHILE.L | — | 3+3×n<br>n: Number of comparison longwords |
| System manipulation instructions | • RTE | — | 6 |
| | • RTFI | — | 3 |

?: Conditional operator

Note 1. floor(x): Max. integer that is smaller than x
Note 2. For the number of cycles for throughput and latency, refer to section 2.8.3, Calculation of the Instruction Processing Time.
Note 3. The PUSHM instruction is converted into multiple store operations. The pipeline processing is the same as the one for the store operations of the MOV instruction, where the operation is repeated for the number of specified registers.
Note 4. The POPM instruction is converted into multiple load operations. The pipeline processing is the same as the one for the load operations of the MOV instruction, where the operation is repeated for the number of specified registers.
Note 5. Each of the SCMPU, SMOVU, SWHILE, and SUNTIL instructions ends the execution regardless of the specified cycles, if the end condition is satisfied during execution.

Figure 2.10 to Figure 2.14 show the operation of instructions that are converted into basic multiple micro-operations.
Note: • mop: Micro-operation, stall: Pipeline stall

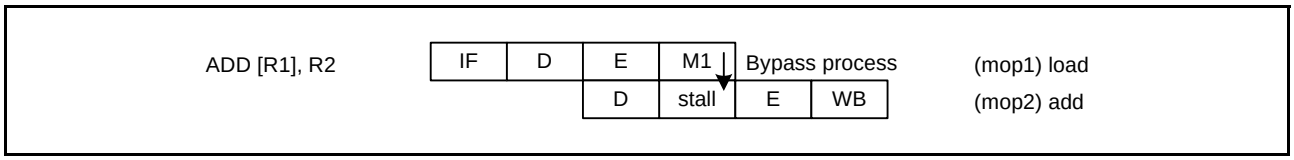



**Figure 2.10 Arithmetic/Logic Instruction (Memory Source Operand)**

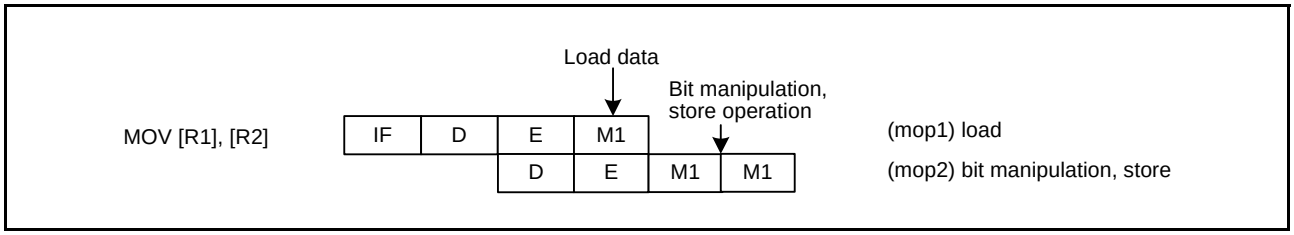



**Figure 2.11 MOV Instruction (Memory-Memory), Bit Manipulation Instruction (Memory Source Operand)**

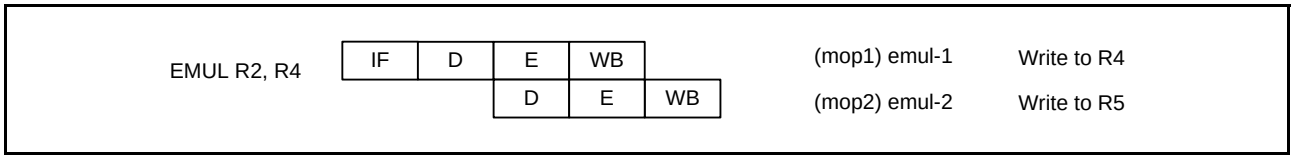



**Figure 2.12 EMUL, EMULU Instructions (Register- Register, Register-Immediate)**

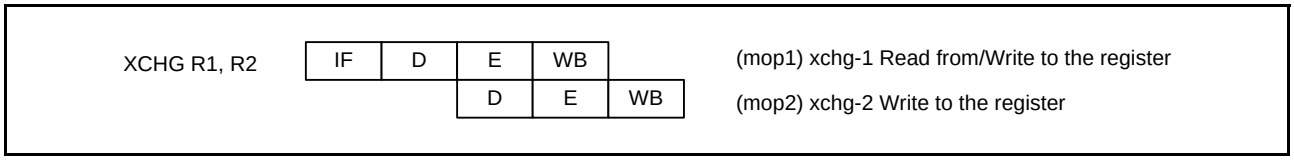



**Figure 2.13 XCHG Instruction (Registers)**

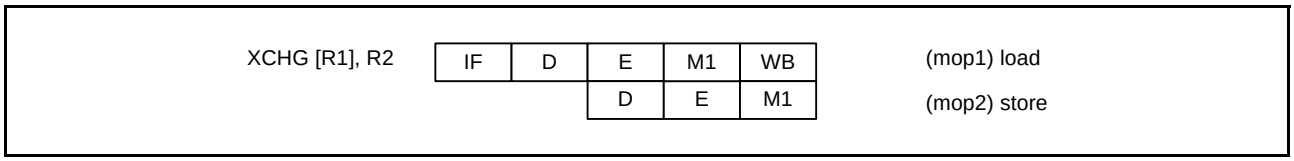



**Figure 2.14 XCHG Instruction (Memory Source Operand)**

### 2.8.2.3 Pipeline Basic Operation

In the ideal pipeline processing, each stage is executed in one cycle, though all instructions may not be pipelined in due to the processing in each stage and the branch execution.
The CPU controls the pipeline stage with the IF stage in the unit of instructions, while the D and subsequent stages in the unit of micro-operations.
The figures below show the pipeline processing of typical cases.
Note: • mop: Micro-operation, stall: Pipeline stall

#### (1) Pipeline Flow with Stalls

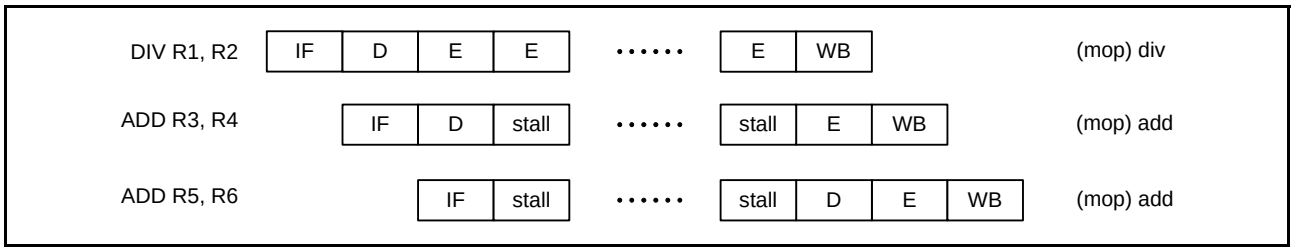



**Figure 2.15 When an Instruction which Requires Multiple Cycles is Executed in the E Stage**

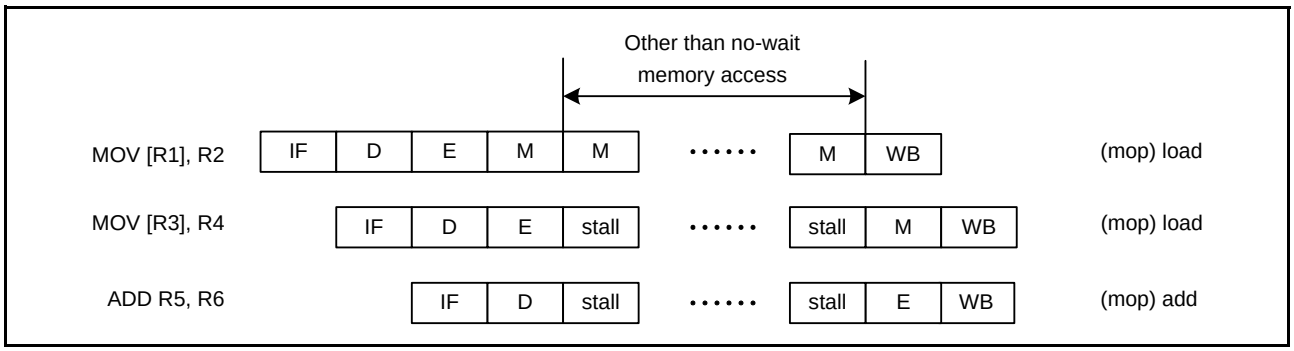



**Figure 2.16 When an Instruction which Requires more than One Cycle for its Operand Access is Executed**

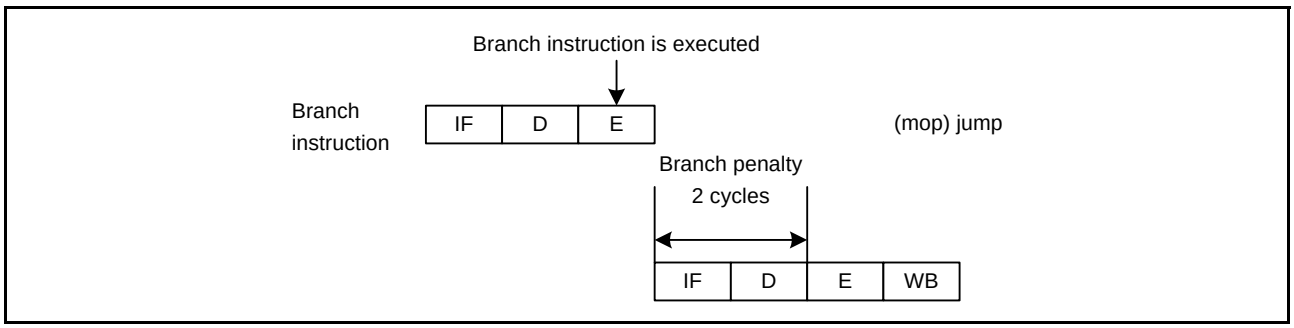



**Figure 2.17 When a Branch Instruction is Executed (an Unconditional Branch Instruction is Executed or the Condition is Satisfied for a Conditional Branch Instruction)**

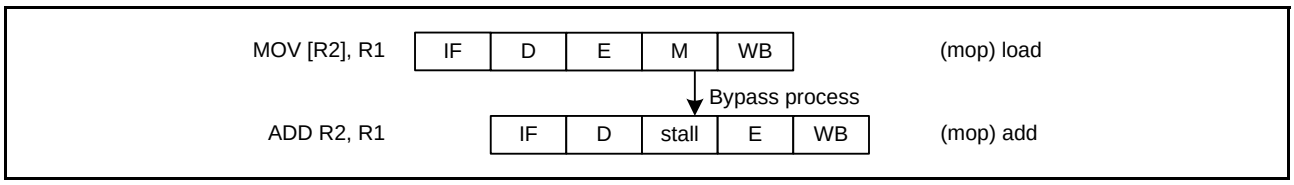



**Figure 2.18 When the Subsequent Instruction Uses an Operand Read from the Memory**

### (2) Pipeline Flow with no Stall

#### (a) Bypass process

Even when the result of the preceding instruction will be used in a subsequent instruction, the operation processing between registers is pipelined in by the bypass process.

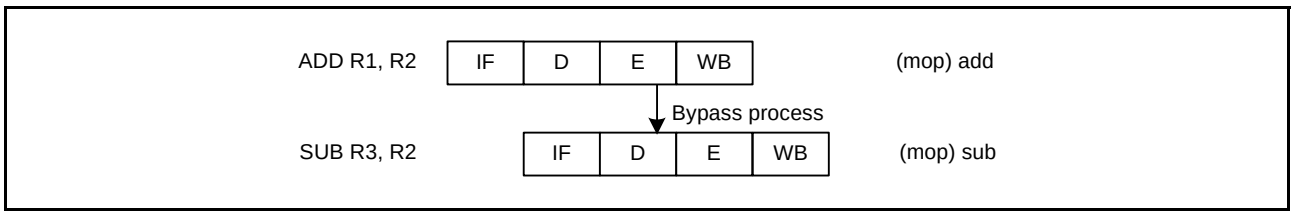



**Figure 2.19 Bypass Process**

#### (b) When WB stages for the memory load and for the operation are overlapped

Even when the WB stages for the memory load and for the operation are overlapped, the operation processing is pipelined in, because the load data and the operation result can be written to the register at the same timing.

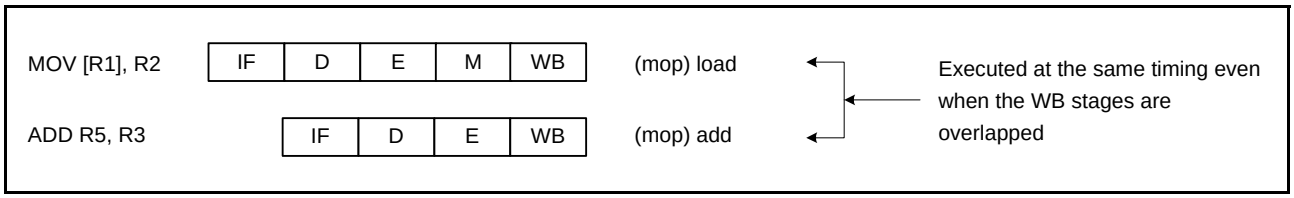



**Figure 2.20 When WB Stages for the Memory Load and for the Operation are Overlapped**

#### (c) When subsequent instruction writes to the same register before the end of memory load

Even when the subsequent instruction writes to the same register before the end of memory load, the operation processing is pipelined in, because the WB stage for the memory load is canceled.

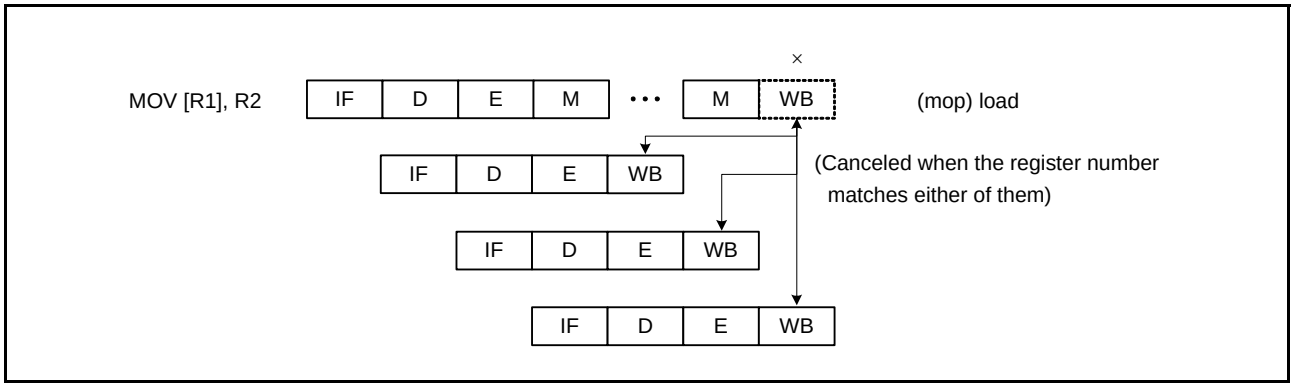



**Figure 2.21 When Subsequent Instruction Writes to the Same Register before the End of Memory Load**

### (d) When the load data is not used by the subsequent instruction

When the load data is not used by the subsequent instruction, the subsequent operations are in fact executed earlier and the operation processing ends (out-of-order completion).

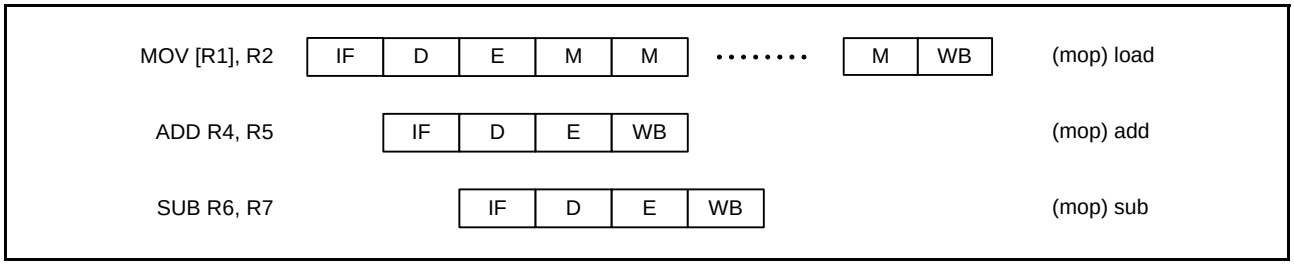



**Figure 2.22 When Load Data is not Used by the Subsequent Instruction**

## 2.8.3 Calculation of the Instruction Processing Time

Though the instruction processing time of the CPU varies according to the pipeline processing, the approximate time can be calculated in the following methods.

- Count the number of cycles (see Table 2.13 and Table 2.14)
- When the load data is used by the subsequent instruction, the number of cycles described as “latency” is counted as the number of cycles for the memory load instruction. For the cycles other than the memory load instruction, the number of cycles described as “throughput” is counted.
- If the instruction fetch stall is generated, the number of cycles increments.
- Depending on the system configuration, multiple cycles are required for the memory access.

### 2.8.4 Numbers of Cycles for Response to Interrupts

Table 2.15 lists numbers of cycles taken by processing for response to interrupts.

**Table 2.15 Numbers of Cycles for Response to Interrupts**

| Type of Interrupt Request/Details of Processing | Fast Interrupt | Other Interrupts |
|---|---|---|
| ICU<br>Judgment of priority order | 2 cycles | |
| CPU<br>Number of cycles from notification to acceptance of the interrupt request | N cycles<br>(varies with the instruction being executed at the time the interrupt was received) | |
| CPU Pre-processing by hardware<br>Saving the current PC and PSW values in RAM<br>(or in control registers in the case of the fast interrupt)<br>Reading of the vector<br>Branching to the start of the exception handling routine | 4 cycles | 6 cycles |

Times calculated from the values in Table 2.15 will be applicable when access to memory from the CPU is processed with no waiting. The ROM and RAM in products of this MCU allow such access. Numbers of cycles for response to interrupts can be minimized by placing program code (and vectors) in ROM and the stack in RAM. Furthermore, place the addresses where the exception handling routine start on 8-byte boundaries.

For information on the number of cycles from notification to acceptance of the interrupt request, indicated by N in the table above, see Table 2.13, Instructions that are Converted into a Single Micro-Operation, and Table 2.14, Instructions that are Converted into Multiple Micro-Operations.

The timing of interrupt acceptance depends on the state of the pipelines. For more information on this, refer to section 13.3.1, Acceptance Timing and Saved PC Value.

# 3. Operating Modes

## 3.1 Operating Mode Types and Selection

Operating modes are selected by the pin level when a reset is released.
Table 3.1 shows the relationship between levels on the mode setting pins (MD and UB#) on release from the reset state and the operating mode selected at that time. For details on each of the operating modes, refer to section 3.3, Details of Operating Modes.

**Table 3.1 Selection of Operating Modes by the Mode Setting Pin**

| Mode Setting Pin | | Operating Mode |
|---|---|---|
| MD *1 | UB# | |
| Low | Low | Boot mode (USB interface mode) |
| Low | High or open | Boot mode (SCI mode) |
| High | — | Single-chip mode |

Note 1. Do not change the level on the MD pin while the MCU is operating.

The endian is selectable in single-chip mode. Endian is set by the MDE.MDE[2:0] bits in the option-setting memory. For the correspondence between the setting and endian, see Table 3.2.

**Table 3.2 Endian Setting in Single-Chip Mode**

| MDE.MDE[2:0] Bits | Endian |
|---|---|
| 000b | Big endian |
| 111b | Little endian |

## 3.2 Register Descriptions

### 3.2.1 Mode Monitor Register (MDMONR)

Address(es): 0008 0000h

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | MD |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | x | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0/1*1 |

Note 1. Depends on the setting of the mode pin (MD). When the MD pin is low, the bit value is 0; otherwise, the bit value is 1.

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b0 | MD | MD Pin Status Flag | 0: The MD pin is low.<br>1: The MD pin is high. | R |
| b7 to b1 | — | Reserved | These bits are read as 0. | R |
| b8 | — | Reserved | The read value is undefined. | R |
| b15 to b9 | — | Reserved | These bits are read as 0. | R |

## 3.2.2 System Control Register 1 (SYSCR1)

Address(es): 0008 0008h

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | RAME |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b0 | RAME | RAM Enable | 0: The RAM is disabled.<br>1: The RAM is enabled. | R/W |
| b15 to b1 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |

Note: •Set the PRCR.PRC1 bit to 1 (write enabled) before rewriting this register.

### RAME Bit (RAM Enable)

The RAME bit enables or disables the RAM.

A 0 should not be written to this bit during access to the RAM. When accessing the RAM immediately after changing the RAME bit from 0 (RAM disabled) to 1 (RAM enabled), make sure that the RAME bit is 1 before the access.

Even when the RAME bit is cleared to 0, the RAM retains its value. To retain the value in the RAM, keep the specified RAM standby voltage (VRAM). For details, refer to section 36, Electrical Characteristics.

## 3.3 Details of Operating Modes

### 3.3.1 Single-Chip Mode

In this mode, all I/O ports can be used as general input/output ports, peripheral function input/output, or interrupt input pins.
The chip starts up in single-chip mode if the high level is on the MD pin on release from the reset state.

### 3.3.2 Boot Mode

In this mode, the on-chip flash memory modifying program (boot program) stored in a dedicated area within the MCU operates. The on-chip flash memory (ROM and E2 DataFlash) can be modified from outside the MCU by using a USB or universal asynchronous receiver/transmitter (SCI1). For details, refer to section 35, Flash Memory.
The chip starts up in boot mode if the low level is on the MD pin on release from the reset state.

#### 3.3.2.1 Boot Mode (USB Interface Mode)

When a reset is released while pins MD and UB# are low, USB interface mode is selected. For details on USB interface mode, refer to section 35.7.2, SCI Mode.

#### 3.3.2.2 Boot Mode (SCI Mode)

When a reset is released while the MD pin is low and the UB# pin is high or left open, SCI mode is selected. For details on SCI mode, refer to section 35.7.2, SCI Mode.

## 3.4 Transitions of Operating Modes

### 3.4.1 Mode Setting Pin Levels and Operating Mode Transitions

Figure 3.1 shows operating mode transitions according to the setting of the MD and UB# pins.

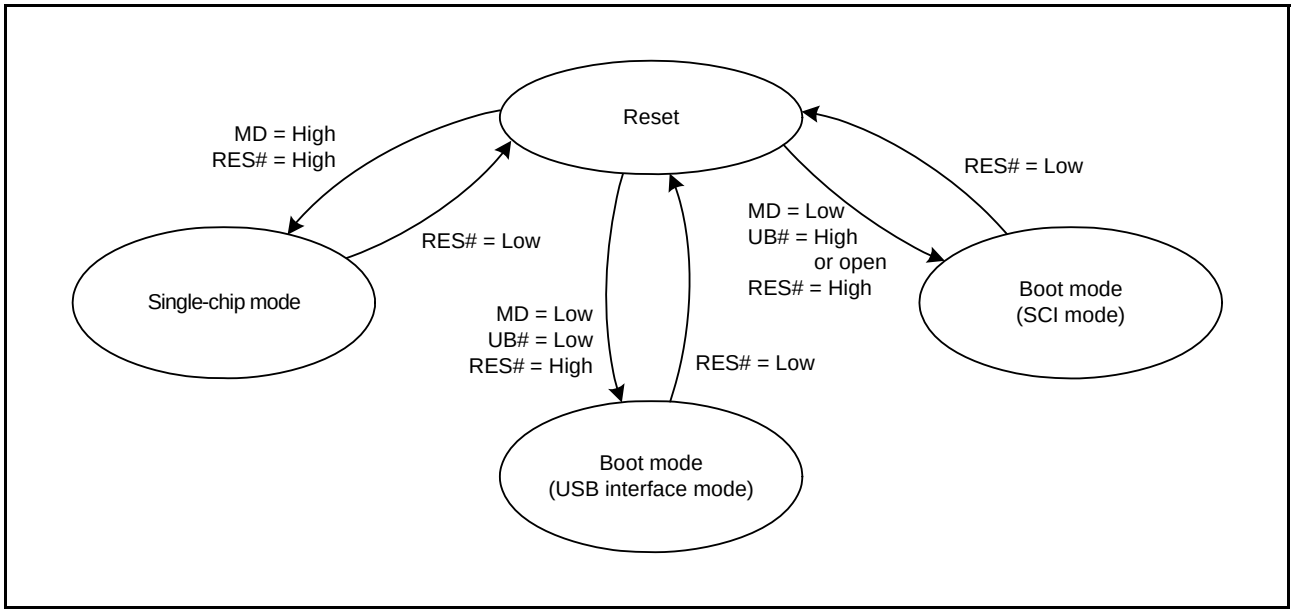
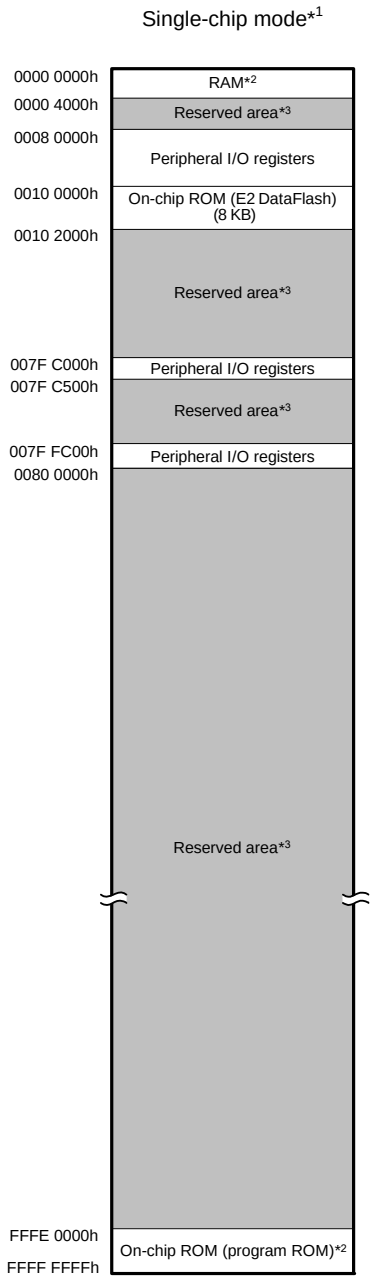



**Figure 3.1 Mode Setting Pin Levels and Operating Modes**

# 4. Address Space

## 4.1 Address Space

This MCU has a 4-Gbyte address space, consisting of the range of addresses from 0000 0000h to FFFF FFFFh. That is, linear access to an address space of up to 4 Gbytes is possible, and this contains both program and data areas.
Figure 4.1 shows the memory map.




Note 1. The address space in boot mode is the same as the address space in single-chip mode.
Note 2. The capacity of ROM/RAM differs depending on the products.

| ROM (bytes) | | RAM (bytes) | |
|---|---|---|---|
| **Capacity** | **Address** | **Capacity** | **Address** |
| 128 K | FFFE 0000h to FFFF FFFFh | 16 K | 0000 0000h to 0000 3FFFh |
| 96 K | FFFE 8000h to FFFF FFFFh | | |
| 64 K | FFFF 0000h to FFFF FFFFh | 10 K | 0000 0000h to 0000 27FFh |
| 32 K | FFFF 8000h to FFFF FFFFh | | |
| 16 K | FFFF C000h to FFFF FFFFh | 8 K | 0000 0000h to 0000 1FFFh |

Note: • See Table 1.3, List of Products, for the product type name.

Note 3. Reserved areas should not be accessed.

**Figure 4.1 Memory Map**

# 5. I/O Registers

This section provides information on the on-chip I/O register addresses and bit configuration. The information is given as shown below. Notes on writing to I/O registers are also given below.

## (1) I/O register addresses (address order)

- Registers are listed from the lower allocation addresses.
- Registers are classified according to module symbols.
- Numbers of cycles for access indicate numbers of cycles of the given base clock.
- Among the internal I/O register area, addresses not listed in the list of registers are reserved. Reserved addresses must not be accessed. Do not access these addresses; otherwise, the operation when accessing these bits and subsequent operations cannot be guaranteed.

## (2) Notes on writing to I/O registers

While writing to an I/O register, the CPU starts executing subsequent instructions before the I/O register write access is completed. This may cause the subsequent instructions to be executed before the write value is reflected in the operation. The examples below show how subsequent instructions must be executed after a write access to an I/O register is completed.

**[Examples of cases requiring special care]**

- The subsequent instruction must be executed while an interrupt request is disabled with the IENj bit in IERn of the ICU (interrupt request enable bit) cleared to 0.
- A WAIT instruction is executed immediately after the preprocessing for causing a transition to the low power consumption state.

In the above cases, after writing to an I/O register, wait until the write operation is completed using the following procedure and then execute the subsequent instruction.

(a) Write to an I/O register.
(b) Read the value in the I/O register and write it to a general register.
(c) Execute the operation using the value read.
(d) Execute the subsequent instruction.

Example of instructions

- Byte-size I/O registers

```
MOV.L  #SFR_ADDR, R1
MOV.B  #SFR_DATA, [R1]
CMP    [R1].UB, R1
;; Next process
```

- Word-size I/O registers

```
MOV.L  #SFR_ADDR, R1
MOV.W  #SFR_DATA, [R1]
CMP    [R1].W, R1
;; Next process
```

- Longword-size I/O registers

```
MOV.L  #SFR_ADDR, R1
MOV.L  #SFR_DATA, [R1]
CMP    [R1].L, R1
;; Next process
```

When executing an instruction after writing to multiple registers, only read the last I/O register written to and execute the instruction using that value; it is not necessary to execute the instruction using the values written to all the registers.

### (3) Number of cycles necessary for accessing I/O registers

See Table 5.1 for details on the number of clock cycles necessary for accessing I/O registers.
The number of access cycles to I/O registers is obtained by following equation.*1

Number of access cycles to I/O registers = Number of bus cycles for internal main bus 1 +
Number of divided clock synchronization cycles +
Number of bus cycles for internal peripheral buses 1 to 6

The number of bus cycles of internal peripheral buses 1 to 6 differs according to the register to be accessed.
When peripheral functions connected to internal peripheral buses 2 to 6 or registers for the external bus control unit (except for bus error related registers) are accessed, the number of divided clock synchronization cycles is added.
The number of divided clock synchronization cycles differs depending on the frequency ratio between ICLK and PCLK (or FCLK) or bus access timing.
In the peripheral function unit, when the frequency ratio of ICLK is equal to or greater than that of PCLK (or FCLK), the sum of the number of bus cycles for internal main bus 1 and the number of the divided clock synchronization cycles will be one cycle of PCLK (or FCLK) at a maximum. Therefore, one PCLK (or FCLK) has been added to the number of access cycles shown in Table 5.1.
When the frequency ratio of ICLK is lower than that of PCLK (or FCLK), the subsequent bus access is started from the ICLK cycle following the completion of the access to the peripheral functions. Therefore, the access cycles are described on an ICLK basis.

Note 1. This applies to the number of cycles when the access from the CPU does not conflict with the instruction fetching to the external memory or bus access from the different bus master (DTC).

### (4) Notes on sleep mode and mode transitions

During sleep mode or mode transitions, do not write to the system control related registers (indicated by ‘SYSTEM’ in the Module Symbol column in Table 5.1, List of I/O Registers (Address Order)).

## 5.1 I/O Register Addresses (Address Order)

**Table 5.1 List of I/O Registers (Address Order) (1/16)**

| Address | Module Symbol | Register Name | Register Symbol | Number of Bits | Access Size | Number of Access States | Reference Section |
|---|---|---|---|---|---|---|---|
| 0008 0000h | SYSTEM | Mode Monitor Register | MDMONR | 16 | 16 | 3 ICLK | section 3. |
| 0008 0008h | SYSTEM | System Control Register 1 | SYSCR1 | 16 | 16 | 3 ICLK | section 3. |
| 0008 000Ch | SYSTEM | Standby Control Register | SBYCR | 16 | 16 | 3 ICLK | section 11. |
| 0008 0010h | SYSTEM | Module Stop Control Register A | MSTPCRA | 32 | 32 | 3 ICLK | section 11. |
| 0008 0014h | SYSTEM | Module Stop Control Register B | MSTPCRB | 32 | 32 | 3 ICLK | section 11. |
| 0008 0018h | SYSTEM | Module Stop Control Register C | MSTPCRC | 32 | 32 | 3 ICLK | section 11. |
| 0008 0020h | SYSTEM | System Clock Control Register | SCKCR | 32 | 32 | 3 ICLK | section 9. |
| 0008 0026h | SYSTEM | System Clock Control Register 3 | SCKCR3 | 16 | 16 | 3 ICLK | section 9. |
| 0008 0028h | SYSTEM | PLL Control Register | PLLCR | 16 | 16 | 3 ICLK | section 9. |
| 0008 002Ah | SYSTEM | PLL Control Register 2 | PLLCR2 | 8 | 8 | 3 ICLK | section 9. |
| 0008 0032h | SYSTEM | Main Clock Oscillator Control Register | MOSCCR | 8 | 8 | 3 ICLK | section 9. |
| 0008 0033h | SYSTEM | Sub-Clock Oscillator Control Register | SOSCCR | 8 | 8 | 3 ICLK | section 9. |
| 0008 0034h | SYSTEM | Low-Speed On-Chip Oscillator Control Register | LOCOCR | 8 | 8 | 3 ICLK | section 9. |
| 0008 0035h | SYSTEM | IWDT-Dedicated On-Chip Oscillator Control Register | ILOCOCR | 8 | 8 | 3 ICLK | section 9. |
| 0008 0036h | SYSTEM | High-Speed On-Chip Oscillator Control Register | HOCOCR | 8 | 8 | 3 ICLK | section 9. |
| 0008 003Ch | SYSTEM | Oscillation Stabilization Flag Register | OSCOVFSR | 8 | 8 | 3 ICLK | section 9. |
| 0008 003Eh | SYSTEM | CLKOUT Output Control Register | CKOCR | 16 | 16 | 3 ICLK | section 9. |
| 0008 0040h | SYSTEM | Oscillation Stop Detection Control Register | OSTDCR | 8 | 8 | 3 ICLK | section 9. |
| 0008 0041h | SYSTEM | Oscillation Stop Detection Status Register | OSTDSR | 8 | 8 | 3 ICLK | section 9. |
| 0008 00A0h | SYSTEM | Operating Power Control Register | OPCCR | 8 | 8 | 3 ICLK | section 11. |
| 0008 00A1h | SYSTEM | Sleep Mode Return Clock Source Switching Register | RSTCKCR | 8 | 8 | 3 ICLK | section 11. |
| 0008 00A2h | SYSTEM | Main Clock Oscillator Wait Control Register | MOSCWTCR | 8 | 8 | 3 ICLK | section 9. |
| 0008 00A5h | SYSTEM | High-Speed On-Chip Oscillator Wait Control Register | HOCOWTCR | 8 | 8 | 3 ICLK | section 9. |
| 0008 00AAh | SYSTEM | Sub Operating Power Control Register | SOPCCR | 8 | 8 | 3 ICLK | section 11. |
| 0008 00C0h | SYSTEM | Reset Status Register 2 | RSTSR2 | 8 | 8 | 3 ICLK | section 6. |
| 0008 00C2h | SYSTEM | Software Reset Register | SWRR | 16 | 16 | 3 ICLK | section 6. |
| 0008 00E0h | SYSTEM | Voltage Monitoring 1 Circuit Control Register 1 | LVD1CR1 | 8 | 8 | 3 ICLK | section 8. |
| 0008 00E1h | SYSTEM | Voltage Monitoring 1 Circuit Status Register | LVD1SR | 8 | 8 | 3 ICLK | section 8. |
| 0008 00E2h | SYSTEM | Voltage Monitoring 2 Circuit Control Register 1 | LVD2CR1 | 8 | 8 | 3 ICLK | section 8. |
| 0008 00E3h | SYSTEM | Voltage Monitoring 2 Circuit Status Register | LVD2SR | 8 | 8 | 3 ICLK | section 8. |
| 0008 03FEh | SYSTEM | Protect Register | PRCR | 16 | 16 | 3 ICLK | section 12. |
| 0008 1300h | BSC | Bus Error Status Clear Register | BERCLR | 8 | 8 | 2 ICLK | section 15. |
| 0008 1304h | BSC | Bus Error Monitoring Enable Register | BEREN | 8 | 8 | 2 ICLK | section 15. |
| 0008 1308h | BSC | Bus Error Status Register 1 | BERSR1 | 8 | 8 | 2 ICLK | section 15. |
| 0008 130Ah | BSC | Bus Error Status Register 2 | BERSR2 | 16 | 16 | 2 ICLK | section 15. |
| 0008 1310h | BSC | Bus Priority Control Register | BUSPRI | 16 | 16 | 2 ICLK | section 15. |
| 0008 2400h | DTC | DTC Control Register | DTCCR | 8 | 8 | 2 ICLK | section 16. |
| 0008 2404h | DTC | DTC Vector Base Register | DTCVBR | 32 | 32 | 2 ICLK | section 16. |
| 0008 2408h | DTC | DTC Address Mode Register | DTCADMOD | 8 | 8 | 2 ICLK | section 16. |
| 0008 240Ch | DTC | DTC Module Start Register | DTCST | 8 | 8 | 2 ICLK | section 16. |
| 0008 240Eh | DTC | DTC Status Register | DTCSTS | 16 | 16 | 2 ICLK | section 16. |
| 0008 7010h | ICU | Interrupt Request Register 016 | IR016 | 8 | 8 | 2 ICLK | section 14. |
| 0008 701Bh | ICU | Interrupt Request Register 027 | IR027 | 8 | 8 | 2 ICLK | section 14. |
| 0008 701Ch | ICU | Interrupt Request Register 028 | IR028 | 8 | 8 | 2 ICLK | section 14. |
| 0008 701Dh | ICU | Interrupt Request Register 029 | IR029 | 8 | 8 | 2 ICLK | section 14. |
| 0008 7020h | ICU | Interrupt Request Register 032 | IR032 | 8 | 8 | 2 ICLK | section 14. |
| 0008 7021h | ICU | Interrupt Request Register 033 | IR033 | 8 | 8 | 2 ICLK | section 14. |
| 0008 7022h | ICU | Interrupt Request Register 034 | IR034 | 8 | 8 | 2 ICLK | section 14. |
| 0008 7024h | ICU | Interrupt Request Register 036 | IR036 | 8 | 8 | 2 ICLK | section 14. |

**Table 5.1 List of I/O Registers (Address Order) (2/16)**

| Address | Module Symbol | Register Name | Register Symbol | Number of Bits | Access Size | Number of Access States | Reference Section |
|---|---|---|---|---|---|---|---|
| 0008 7025h | ICU | Interrupt Request Register 037 | IR037 | 8 | 8 | 2 ICLK | section 14. |
| 0008 7026h | ICU | Interrupt Request Register 038 | IR038 | 8 | 8 | 2 ICLK | section 14. |
| 0008 702Ch | ICU | Interrupt Request Register 044 | IR044 | 8 | 8 | 2 ICLK | section 14. |
| 0008 702Dh | ICU | Interrupt Request Register 045 | IR045 | 8 | 8 | 2 ICLK | section 14. |
| 0008 702Eh | ICU | Interrupt Request Register 046 | IR046 | 8 | 8 | 2 ICLK | section 14. |
| 0008 702Fh | ICU | Interrupt Request Register 047 | IR047 | 8 | 8 | 2 ICLK | section 14. |
| 0008 7039h | ICU | Interrupt Request Register 057 | IR057 | 8 | 8 | 2 ICLK | section 14. |
| 0008 703Fh | ICU | Interrupt Request Register 063 | IR063 | 8 | 8 | 2 ICLK | section 14. |
| 0008 7040h | ICU | Interrupt Request Register 064 | IR064 | 8 | 8 | 2 ICLK | section 14. |
| 0008 7041h | ICU | Interrupt Request Register 065 | IR065 | 8 | 8 | 2 ICLK | section 14. |
| 0008 7042h | ICU | Interrupt Request Register 066 | IR066 | 8 | 8 | 2 ICLK | section 14. |
| 0008 7043h | ICU | Interrupt Request Register 067 | IR067 | 8 | 8 | 2 ICLK | section 14. |
| 0008 7044h | ICU | Interrupt Request Register 068 | IR068 | 8 | 8 | 2 ICLK | section 14. |
| 0008 7045h | ICU | Interrupt Request Register 069 | IR069 | 8 | 8 | 2 ICLK | section 14. |
| 0008 7046h | ICU | Interrupt Request Register 070 | IR070 | 8 | 8 | 2 ICLK | section 14. |
| 0008 7047h | ICU | Interrupt Request Register 071 | IR071 | 8 | 8 | 2 ICLK | section 14. |
| 0008 7058h | ICU | Interrupt Request Register 088 | IR088 | 8 | 8 | 2 ICLK | section 14. |
| 0008 7059h | ICU | Interrupt Request Register 089 | IR089 | 8 | 8 | 2 ICLK | section 14. |
| 0008 705Ah | ICU | Interrupt Request Register 090 | IR090 | 8 | 8 | 2 ICLK | section 14. |
| 0008 705Ch | ICU | Interrupt Request Register 092 | IR092 | 8 | 8 | 2 ICLK | section 14. |
| 0008 705Dh | ICU | Interrupt Request Register 093 | IR093 | 8 | 8 | 2 ICLK | section 14. |
| 0008 7066h | ICU | Interrupt Request Register 102 | IR102 | 8 | 8 | 2 ICLK | section 14. |
| 0008 7067h | ICU | Interrupt Request Register 103 | IR103 | 8 | 8 | 2 ICLK | section 14. |
| 0008 706Ah | ICU | Interrupt Request Register 106 | IR106 | 8 | 8 | 2 ICLK | section 14. |
| 0008 7072h | ICU | Interrupt Request Register 114 | IR114 | 8 | 8 | 2 ICLK | section 14. |
| 0008 7073h | ICU | Interrupt Request Register 115 | IR115 | 8 | 8 | 2 ICLK | section 14. |
| 0008 7074h | ICU | Interrupt Request Register 116 | IR116 | 8 | 8 | 2 ICLK | section 14. |
| 0008 7075h | ICU | Interrupt Request Register 117 | IR117 | 8 | 8 | 2 ICLK | section 14. |
| 0008 7076h | ICU | Interrupt Request Register 118 | IR118 | 8 | 8 | 2 ICLK | section 14. |
| 0008 7077h | ICU | Interrupt Request Register 119 | IR119 | 8 | 8 | 2 ICLK | section 14. |
| 0008 7078h | ICU | Interrupt Request Register 120 | IR120 | 8 | 8 | 2 ICLK | section 14. |
| 0008 7079h | ICU | Interrupt Request Register 121 | IR121 | 8 | 8 | 2 ICLK | section 14. |
| 0008 707Ah | ICU | Interrupt Request Register 122 | IR122 | 8 | 8 | 2 ICLK | section 14. |
| 0008 707Bh | ICU | Interrupt Request Register 123 | IR123 | 8 | 8 | 2 ICLK | section 14. |
| 0008 707Ch | ICU | Interrupt Request Register 124 | IR124 | 8 | 8 | 2 ICLK | section 14. |
| 0008 707Dh | ICU | Interrupt Request Register 125 | IR125 | 8 | 8 | 2 ICLK | section 14. |
| 0008 707Eh | ICU | Interrupt Request Register 126 | IR126 | 8 | 8 | 2 ICLK | section 14. |
| 0008 707Fh | ICU | Interrupt Request Register 127 | IR127 | 8 | 8 | 2 ICLK | section 14. |
| 0008 7080h | ICU | Interrupt Request Register 128 | IR128 | 8 | 8 | 2 ICLK | section 14. |
| 0008 7081h | ICU | Interrupt Request Register 129 | IR129 | 8 | 8 | 2 ICLK | section 14. |
| 0008 7082h | ICU | Interrupt Request Register 130 | IR130 | 8 | 8 | 2 ICLK | section 14. |
| 0008 7083h | ICU | Interrupt Request Register 131 | IR131 | 8 | 8 | 2 ICLK | section 14. |
| 0008 7084h | ICU | Interrupt Request Register 132 | IR132 | 8 | 8 | 2 ICLK | section 14. |
| 0008 7085h | ICU | Interrupt Request Register 133 | IR133 | 8 | 8 | 2 ICLK | section 14. |
| 0008 7086h | ICU | Interrupt Request Register 134 | IR134 | 8 | 8 | 2 ICLK | section 14. |
| 0008 7087h | ICU | Interrupt Request Register 135 | IR135 | 8 | 8 | 2 ICLK | section 14. |
| 0008 7088h | ICU | Interrupt Request Register 136 | IR136 | 8 | 8 | 2 ICLK | section 14. |
| 0008 7089h | ICU | Interrupt Request Register 137 | IR137 | 8 | 8 | 2 ICLK | section 14. |
| 0008 708Ah | ICU | Interrupt Request Register 138 | IR138 | 8 | 8 | 2 ICLK | section 14. |
| 0008 708Bh | ICU | Interrupt Request Register 139 | IR139 | 8 | 8 | 2 ICLK | section 14. |
| 0008 708Ch | ICU | Interrupt Request Register 140 | IR140 | 8 | 8 | 2 ICLK | section 14. |
| 0008 708Dh | ICU | Interrupt Request Register 141 | IR141 | 8 | 8 | 2 ICLK | section 14. |

**Table 5.1 List of I/O Registers (Address Order) (3/16)**

| Address | Module Symbol | Register Name | Register Symbol | Number of Bits | Access Size | Number of Access States | Reference Section |
|---|---|---|---|---|---|---|---|
| 0008 70AAh | ICU | Interrupt Request Register 170 | IR170 | 8 | 8 | 2 ICLK | section 14. |
| 0008 70ABh | ICU | Interrupt Request Register 171 | IR171 | 8 | 8 | 2 ICLK | section 14. |
| 0008 70DAh | ICU | Interrupt Request Register 218 | IR218 | 8 | 8 | 2 ICLK | section 14. |
| 0008 70DBh | ICU | Interrupt Request Register 219 | IR219 | 8 | 8 | 2 ICLK | section 14. |
| 0008 70DCh | ICU | Interrupt Request Register 220 | IR220 | 8 | 8 | 2 ICLK | section 14. |
| 0008 70DDh | ICU | Interrupt Request Register 221 | IR221 | 8 | 8 | 2 ICLK | section 14. |
| 0008 70DEh | ICU | Interrupt Request Register 222 | IR222 | 8 | 8 | 2 ICLK | section 14. |
| 0008 70DFh | ICU | Interrupt Request Register 223 | IR223 | 8 | 8 | 2 ICLK | section 14. |
| 0008 70E0h | ICU | Interrupt Request Register 224 | IR224 | 8 | 8 | 2 ICLK | section 14. |
| 0008 70E1h | ICU | Interrupt Request Register 225 | IR225 | 8 | 8 | 2 ICLK | section 14. |
| 0008 70EEh | ICU | Interrupt Request Register 238 | IR238 | 8 | 8 | 2 ICLK | section 14. |
| 0008 70EFh | ICU | Interrupt Request Register 239 | IR239 | 8 | 8 | 2 ICLK | section 14. |
| 0008 70F0h | ICU | Interrupt Request Register 240 | IR240 | 8 | 8 | 2 ICLK | section 14. |
| 0008 70F1h | ICU | Interrupt Request Register 241 | IR241 | 8 | 8 | 2 ICLK | section 14. |
| 0008 70F2h | ICU | Interrupt Request Register 242 | IR242 | 8 | 8 | 2 ICLK | section 14. |
| 0008 70F3h | ICU | Interrupt Request Register 243 | IR243 | 8 | 8 | 2 ICLK | section 14. |
| 0008 70F4h | ICU | Interrupt Request Register 244 | IR244 | 8 | 8 | 2 ICLK | section 14. |
| 0008 70F5h | ICU | Interrupt Request Register 245 | IR245 | 8 | 8 | 2 ICLK | section 14. |
| 0008 70F6h | ICU | Interrupt Request Register 246 | IR246 | 8 | 8 | 2 ICLK | section 14. |
| 0008 70F7h | ICU | Interrupt Request Register 247 | IR247 | 8 | 8 | 2 ICLK | section 14. |
| 0008 70F8h | ICU | Interrupt Request Register 248 | IR248 | 8 | 8 | 2 ICLK | section 14. |
| 0008 70F9h | ICU | Interrupt Request Register 249 | IR249 | 8 | 8 | 2 ICLK | section 14. |
| 0008 711Bh | ICU | DTC Activation Enable Register 027 | DTCER027 | 8 | 8 | 2 ICLK | section 14. |
| 0008 711Ch | ICU | DTC Activation Enable Register 028 | DTCER028 | 8 | 8 | 2 ICLK | section 14. |
| 0008 711Dh | ICU | DTC Activation Enable Register 029 | DTCER029 | 8 | 8 | 2 ICLK | section 14. |
| 0008 7124h | ICU | DTC Activation Enable Register 036 | DTCER036 | 8 | 8 | 2 ICLK | section 14. |
| 0008 7125h | ICU | DTC Activation Enable Register 037 | DTCER037 | 8 | 8 | 2 ICLK | section 14. |
| 0008 712Dh | ICU | DTC Activation Enable Register 045 | DTCER045 | 8 | 8 | 2 ICLK | section 14. |
| 0008 712Eh | ICU | DTC Activation Enable Register 046 | DTCER046 | 8 | 8 | 2 ICLK | section 14. |
| 0008 7140h | ICU | DTC Activation Enable Register 064 | DTCER064 | 8 | 8 | 2 ICLK | section 14. |
| 0008 7141h | ICU | DTC Activation Enable Register 065 | DTCER065 | 8 | 8 | 2 ICLK | section 14. |
| 0008 7142h | ICU | DTC Activation Enable Register 066 | DTCER066 | 8 | 8 | 2 ICLK | section 14. |
| 0008 7143h | ICU | DTC Activation Enable Register 067 | DTCER067 | 8 | 8 | 2 ICLK | section 14. |
| 0008 7144h | ICU | DTC Activation Enable Register 068 | DTCER068 | 8 | 8 | 2 ICLK | section 14. |
| 0008 7145h | ICU | DTC Activation Enable Register 069 | DTCER069 | 8 | 8 | 2 ICLK | section 14. |
| 0008 7146h | ICU | DTC Activation Enable Register 070 | DTCER070 | 8 | 8 | 2 ICLK | section 14. |
| 0008 7147h | ICU | DTC Activation Enable Register 071 | DTCER071 | 8 | 8 | 2 ICLK | section 14. |
| 0008 7166h | ICU | DTC Activation Enable Register 102 | DTCER102 | 8 | 8 | 2 ICLK | section 14. |
| 0008 7167h | ICU | DTC Activation Enable Register 103 | DTCER103 | 8 | 8 | 2 ICLK | section 14. |
| 0008 716Ah | ICU | DTC Activation Enable Register 106 | DTCER106 | 8 | 8 | 2 ICLK | section 14. |
| 0008 7172h | ICU | DTC Activation Enable Register 114 | DTCER114 | 8 | 8 | 2 ICLK | section 14. |
| 0008 7173h | ICU | DTC Activation Enable Register 115 | DTCER115 | 8 | 8 | 2 ICLK | section 14. |
| 0008 7174h | ICU | DTC Activation Enable Register 116 | DTCER116 | 8 | 8 | 2 ICLK | section 14. |
| 0008 7175h | ICU | DTC Activation Enable Register 117 | DTCER117 | 8 | 8 | 2 ICLK | section 14. |
| 0008 7179h | ICU | DTC Activation Enable Register 121 | DTCER121 | 8 | 8 | 2 ICLK | section 14. |
| 0008 717Ah | ICU | DTC Activation Enable Register 122 | DTCER122 | 8 | 8 | 2 ICLK | section 14. |
| 0008 717Dh | ICU | DTC Activation Enable Register 125 | DTCER125 | 8 | 8 | 2 ICLK | section 14. |
| 0008 717Eh | ICU | DTC Activation Enable Register 126 | DTCER126 | 8 | 8 | 2 ICLK | section 14. |
| 0008 7181h | ICU | DTC Activation Enable Register 129 | DTCER129 | 8 | 8 | 2 ICLK | section 14. |
| 0008 7182h | ICU | DTC Activation Enable Register 130 | DTCER130 | 8 | 8 | 2 ICLK | section 14. |
| 0008 7183h | ICU | DTC Activation Enable Register 131 | DTCER131 | 8 | 8 | 2 ICLK | section 14. |
| 0008 7184h | ICU | DTC Activation Enable Register 132 | DTCER132 | 8 | 8 | 2 ICLK | section 14. |

**Table 5.1 List of I/O Registers (Address Order) (4/16)**

| Address | Module Symbol | Register Name | Register Symbol | Number of Bits | Access Size | Number of Access States | Reference Section |
|---|---|---|---|---|---|---|---|
| 0008 7186h | ICU | DTC Activation Enable Register 134 | DTCER134 | 8 | 8 | 2 ICLK | section 14. |
| 0008 7187h | ICU | DTC Activation Enable Register 135 | DTCER135 | 8 | 8 | 2 ICLK | section 14. |
| 0008 7188h | ICU | DTC Activation Enable Register 136 | DTCER136 | 8 | 8 | 2 ICLK | section 14. |
| 0008 7189h | ICU | DTC Activation Enable Register 137 | DTCER137 | 8 | 8 | 2 ICLK | section 14. |
| 0008 718Ah | ICU | DTC Activation Enable Register 138 | DTCER138 | 8 | 8 | 2 ICLK | section 14. |
| 0008 718Bh | ICU | DTC Activation Enable Register 139 | DTCER139 | 8 | 8 | 2 ICLK | section 14. |
| 0008 718Ch | ICU | DTC Activation Enable Register 140 | DTCER140 | 8 | 8 | 2 ICLK | section 14. |
| 0008 718Dh | ICU | DTC Activation Enable Register 141 | DTCER141 | 8 | 8 | 2 ICLK | section 14. |
| 0008 71DBh | ICU | DTC Activation Enable Register 219 | DTCER219 | 8 | 8 | 2 ICLK | section 14. |
| 0008 71DCh | ICU | DTC Activation Enable Register 220 | DTCER220 | 8 | 8 | 2 ICLK | section 14. |
| 0008 71DFh | ICU | DTC Activation Enable Register 223 | DTCER223 | 8 | 8 | 2 ICLK | section 14. |
| 0008 71E0h | ICU | DTC Activation Enable Register 224 | DTCER224 | 8 | 8 | 2 ICLK | section 14. |
| 0008 71EFh | ICU | DTC Activation Enable Register 239 | DTCER239 | 8 | 8 | 2 ICLK | section 14. |
| 0008 71F0h | ICU | DTC Activation Enable Register 240 | DTCER240 | 8 | 8 | 2 ICLK | section 14. |
| 0008 71F7h | ICU | DTC Activation Enable Register 247 | DTCER247 | 8 | 8 | 2 ICLK | section 14. |
| 0008 71F8h | ICU | DTC Activation Enable Register 248 | DTCER248 | 8 | 8 | 2 ICLK | section 14. |
| 0008 7202h | ICU | Interrupt Request Enable Register 02 | IER02 | 8 | 8 | 2 ICLK | section 14. |
| 0008 7203h | ICU | Interrupt Request Enable Register 03 | IER03 | 8 | 8 | 2 ICLK | section 14. |
| 0008 7204h | ICU | Interrupt Request Enable Register 04 | IER04 | 8 | 8 | 2 ICLK | section 14. |
| 0008 7205h | ICU | Interrupt Request Enable Register 05 | IER05 | 8 | 8 | 2 ICLK | section 14. |
| 0008 7207h | ICU | Interrupt Request Enable Register 07 | IER07 | 8 | 8 | 2 ICLK | section 14. |
| 0008 7208h | ICU | Interrupt Request Enable Register 08 | IER08 | 8 | 8 | 2 ICLK | section 14. |
| 0008 720Bh | ICU | Interrupt Request Enable Register 0B | IER0B | 8 | 8 | 2 ICLK | section 14. |
| 0008 720Ch | ICU | Interrupt Request Enable Register 0C | IER0C | 8 | 8 | 2 ICLK | section 14. |
| 0008 720Dh | ICU | Interrupt Request Enable Register 0D | IER0D | 8 | 8 | 2 ICLK | section 14. |
| 0008 720Eh | ICU | Interrupt Request Enable Register 0E | IER0E | 8 | 8 | 2 ICLK | section 14. |
| 0008 720Fh | ICU | Interrupt Request Enable Register 0F | IER0F | 8 | 8 | 2 ICLK | section 14. |
| 0008 7210h | ICU | Interrupt Request Enable Register 10 | IER10 | 8 | 8 | 2 ICLK | section 14. |
| 0008 7211h | ICU | Interrupt Request Enable Register 11 | IER11 | 8 | 8 | 2 ICLK | section 14. |
| 0008 7215h | ICU | Interrupt Request Enable Register 15 | IER15 | 8 | 8 | 2 ICLK | section 14. |
| 0008 721Bh | ICU | Interrupt Request Enable Register 1B | IER1B | 8 | 8 | 2 ICLK | section 14. |
| 0008 721Ch | ICU | Interrupt Request Enable Register 1C | IER1C | 8 | 8 | 2 ICLK | section 14. |
| 0008 721Dh | ICU | Interrupt Request Enable Register 1D | IER1D | 8 | 8 | 2 ICLK | section 14. |
| 0008 721Eh | ICU | Interrupt Request Enable Register 1E | IER1E | 8 | 8 | 2 ICLK | section 14. |
| 0008 721Fh | ICU | Interrupt Request Enable Register 1F | IER1F | 8 | 8 | 2 ICLK | section 14. |
| 0008 72E0h | ICU | Software Interrupt Activation Register | SWINTR | 8 | 8 | 2 ICLK | section 14. |
| 0008 72F0h | ICU | Fast Interrupt Set Register | FIR | 16 | 16 | 2 ICLK | section 14. |
| 0008 7300h | ICU | Interrupt Source Priority Register 000 | IPR000 | 8 | 8 | 2 ICLK | section 14. |
| 0008 7303h | ICU | Interrupt Source Priority Register 003 | IPR003 | 8 | 8 | 2 ICLK | section 14. |
| 0008 7304h | ICU | Interrupt Source Priority Register 004 | IPR004 | 8 | 8 | 2 ICLK | section 14. |
| 0008 7305h | ICU | Interrupt Source Priority Register 005 | IPR005 | 8 | 8 | 2 ICLK | section 14. |
| 0008 7320h | ICU | Interrupt Source Priority Register 032 | IPR032 | 8 | 8 | 2 ICLK | section 14. |
| 0008 7321h | ICU | Interrupt Source Priority Register 033 | IPR033 | 8 | 8 | 2 ICLK | section 14. |
| 0008 7322h | ICU | Interrupt Source Priority Register 034 | IPR034 | 8 | 8 | 2 ICLK | section 14. |
| 0008 7324h | ICU | Interrupt Source Priority Register 036 | IPR036 | 8 | 8 | 2 ICLK | section 14. |
| 0008 7325h | ICU | Interrupt Source Priority Register 037 | IPR037 | 8 | 8 | 2 ICLK | section 14. |
| 0008 7326h | ICU | Interrupt Source Priority Register 038 | IPR038 | 8 | 8 | 2 ICLK | section 14. |
| 0008 732Ch | ICU | Interrupt Source Priority Register 044 | IPR044 | 8 | 8 | 2 ICLK | section 14. |
| 0008 7339h | ICU | Interrupt Source Priority Register 057 | IPR057 | 8 | 8 | 2 ICLK | section 14. |
| 0008 733Fh | ICU | Interrupt Source Priority Register 063 | IPR063 | 8 | 8 | 2 ICLK | section 14. |
| 0008 7340h | ICU | Interrupt Source Priority Register 064 | IPR064 | 8 | 8 | 2 ICLK | section 14. |
| 0008 7341h | ICU | Interrupt Source Priority Register 065 | IPR065 | 8 | 8 | 2 ICLK | section 14. |

**Table 5.1 List of I/O Registers (Address Order) (5/16)**

| Address | Module Symbol | Register Name | Register Symbol | Number of Bits | Access Size | Number of Access States | Reference Section |
|---|---|---|---|---|---|---|---|
| 0008 7342h | ICU | Interrupt Source Priority Register 066 | IPR066 | 8 | 8 | 2 ICLK | section 14. |
| 0008 7343h | ICU | Interrupt Source Priority Register 067 | IPR067 | 8 | 8 | 2 ICLK | section 14. |
| 0008 7344h | ICU | Interrupt Source Priority Register 068 | IPR068 | 8 | 8 | 2 ICLK | section 14. |
| 0008 7345h | ICU | Interrupt Source Priority Register 069 | IPR069 | 8 | 8 | 2 ICLK | section 14. |
| 0008 7346h | ICU | Interrupt Source Priority Register 070 | IPR070 | 8 | 8 | 2 ICLK | section 14. |
| 0008 7347h | ICU | Interrupt Source Priority Register 071 | IPR071 | 8 | 8 | 2 ICLK | section 14. |
| 0008 7358h | ICU | Interrupt Source Priority Register 088 | IPR088 | 8 | 8 | 2 ICLK | section 14. |
| 0008 7359h | ICU | Interrupt Source Priority Register 089 | IPR089 | 8 | 8 | 2 ICLK | section 14. |
| 0008 735Ah | ICU | Interrupt Source Priority Register 090 | IPR090 | 8 | 8 | 2 ICLK | section 14. |
| 0008 735Ch | ICU | Interrupt Source Priority Register 092 | IPR092 | 8 | 8 | 2 ICLK | section 14. |
| 0008 735Dh | ICU | Interrupt Source Priority Register 093 | IPR093 | 8 | 8 | 2 ICLK | section 14. |
| 0008 7366h | ICU | Interrupt Source Priority Register 102 | IPR102 | 8 | 8 | 2 ICLK | section 14. |
| 0008 7367h | ICU | Interrupt Source Priority Register 103 | IPR103 | 8 | 8 | 2 ICLK | section 14. |
| 0008 736Ah | ICU | Interrupt Source Priority Register 106 | IPR106 | 8 | 8 | 2 ICLK | section 14. |
| 0008 7372h | ICU | Interrupt Source Priority Register 114 | IPR114 | 8 | 8 | 2 ICLK | section 14. |
| 0008 7376h | ICU | Interrupt Source Priority Register 118 | IPR118 | 8 | 8 | 2 ICLK | section 14. |
| 0008 7379h | ICU | Interrupt Source Priority Register 121 | IPR121 | 8 | 8 | 2 ICLK | section 14. |
| 0008 737Bh | ICU | Interrupt Source Priority Register 123 | IPR123 | 8 | 8 | 2 ICLK | section 14. |
| 0008 737Dh | ICU | Interrupt Source Priority Register 125 | IPR125 | 8 | 8 | 2 ICLK | section 14. |
| 0008 737Fh | ICU | Interrupt Source Priority Register 127 | IPR127 | 8 | 8 | 2 ICLK | section 14. |
| 0008 7381h | ICU | Interrupt Source Priority Register 129 | IPR129 | 8 | 8 | 2 ICLK | section 14. |
| 0008 7385h | ICU | Interrupt Source Priority Register 133 | IPR133 | 8 | 8 | 2 ICLK | section 14. |
| 0008 7386h | ICU | Interrupt Source Priority Register 134 | IPR134 | 8 | 8 | 2 ICLK | section 14. |
| 0008 738Ah | ICU | Interrupt Source Priority Register 138 | IPR138 | 8 | 8 | 2 ICLK | section 14. |
| 0008 738Bh | ICU | Interrupt Source Priority Register 139 | IPR139 | 8 | 8 | 2 ICLK | section 14. |
| 0008 73AAh | ICU | Interrupt Source Priority Register 170 | IPR170 | 8 | 8 | 2 ICLK | section 14. |
| 0008 73ABh | ICU | Interrupt Source Priority Register 171 | IPR171 | 8 | 8 | 2 ICLK | section 14. |
| 0008 73DAh | ICU | Interrupt Source Priority Register 218 | IPR218 | 8 | 8 | 2 ICLK | section 14. |
| 0008 73DEh | ICU | Interrupt Source Priority Register 222 | IPR222 | 8 | 8 | 2 ICLK | section 14. |
| 0008 73EEh | ICU | Interrupt Source Priority Register 238 | IPR238 | 8 | 8 | 2 ICLK | section 14. |
| 0008 73F2h | ICU | Interrupt Source Priority Register 242 | IPR242 | 8 | 8 | 2 ICLK | section 14. |
| 0008 73F3h | ICU | Interrupt Source Priority Register 243 | IPR243 | 8 | 8 | 2 ICLK | section 14. |
| 0008 73F4h | ICU | Interrupt Source Priority Register 244 | IPR244 | 8 | 8 | 2 ICLK | section 14. |
| 0008 73F5h | ICU | Interrupt Source Priority Register 245 | IPR245 | 8 | 8 | 2 ICLK | section 14. |
| 0008 73F6h | ICU | Interrupt Source Priority Register 246 | IPR246 | 8 | 8 | 2 ICLK | section 14. |
| 0008 73F7h | ICU | Interrupt Source Priority Register 247 | IPR247 | 8 | 8 | 2 ICLK | section 14. |
| 0008 73F8h | ICU | Interrupt Source Priority Register 248 | IPR248 | 8 | 8 | 2 ICLK | section 14. |
| 0008 73F9h | ICU | Interrupt Source Priority Register 249 | IPR249 | 8 | 8 | 2 ICLK | section 14. |
| 0008 7500h | ICU | IRQ Control Register 0 | IRQCR0 | 8 | 8 | 2 ICLK | section 14. |
| 0008 7501h | ICU | IRQ Control Register 1 | IRQCR1 | 8 | 8 | 2 ICLK | section 14. |
| 0008 7502h | ICU | IRQ Control Register 2 | IRQCR2 | 8 | 8 | 2 ICLK | section 14. |
| 0008 7503h | ICU | IRQ Control Register 3 | IRQCR3 | 8 | 8 | 2 ICLK | section 14. |
| 0008 7504h | ICU | IRQ Control Register 4 | IRQCR4 | 8 | 8 | 2 ICLK | section 14. |
| 0008 7505h | ICU | IRQ Control Register 5 | IRQCR5 | 8 | 8 | 2 ICLK | section 14. |
| 0008 7506h | ICU | IRQ Control Register 6 | IRQCR6 | 8 | 8 | 2 ICLK | section 14. |
| 0008 7507h | ICU | IRQ Control Register 7 | IRQCR7 | 8 | 8 | 2 ICLK | section 14. |
| 0008 7510h | ICU | IRQ Pin Digital Filter Enable Register 0 | IRQFLTE0 | 8 | 8 | 2 ICLK | section 14. |
| 0008 7514h | ICU | IRQ Pin Digital Filter Setting Register 0 | IRQFLTC0 | 16 | 16 | 2 ICLK | section 14. |
| 0008 7580h | ICU | Non-Maskable Interrupt Status Register | NMISR | 8 | 8 | 2 ICLK | section 14. |
| 0008 7581h | ICU | Non-Maskable Interrupt Enable Register | NMIER | 8 | 8 | 2 ICLK | section 14. |
| 0008 7582h | ICU | Non-Maskable Interrupt Status Clear Register | NMICLR | 8 | 8 | 2 ICLK | section 14. |
| 0008 7583h | ICU | NMI Pin Interrupt Control Register | NMICR | 8 | 8 | 2 ICLK | section 14. |

**Table 5.1 List of I/O Registers (Address Order) (6/16)**

| Address | Module Symbol | Register Name | Register Symbol | Number of Bits | Access Size | Number of Access States | Reference Section |
|---|---|---|---|---|---|---|---|
| 0008 7590h | ICU | NMI Pin Digital Filter Enable Register | NMIFLTE | 8 | 8 | 2 ICLK | section 14. |
| 0008 7594h | ICU | NMI Pin Digital Filter Setting Register | NMIFLTC | 8 | 8 | 2 ICLK | section 14. |
| 0008 8000h | CMT | Compare Match Timer Start Register 0 | CMSTR0 | 16 | 16 | 2 or 3 PCLKB | section 22. |
| 0008 8002h | CMT0 | Compare Match Timer Control Register | CMCR | 16 | 16 | 2 or 3 PCLKB | section 22. |
| 0008 8004h | CMT0 | Compare Match Timer Counter | CMCNT | 16 | 16 | 2 or 3 PCLKB | section 22. |
| 0008 8006h | CMT0 | Compare Match Timer Constant Register | CMCOR | 16 | 16 | 2 or 3 PCLKB | section 22. |
| 0008 8008h | CMT1 | Compare Match Timer Control Register | CMCR | 16 | 16 | 2 or 3 PCLKB | section 22. |
| 0008 800Ah | CMT1 | Compare Match Timer Counter | CMCNT | 16 | 16 | 2 or 3 PCLKB | section 22. |
| 0008 800Ch | CMT1 | Compare Match Timer Constant Register | CMCOR | 16 | 16 | 2 or 3 PCLKB | section 22. |
| 0008 8030h | IWDT | IWDT Refresh Register | IWDTRR | 8 | 8 | 2 or 3 PCLKB | section 24. |
| 0008 8032h | IWDT | IWDT Control Register | IWDTCR | 16 | 16 | 2 or 3 PCLKB | section 24. |
| 0008 8034h | IWDT | IWDT Status Register | IWDTSR | 16 | 16 | 2 or 3 PCLKB | section 24. |
| 0008 8036h | IWDT | IWDT Reset Control Register | IWDTRCR | 8 | 8 | 2 or 3 PCLKB | section 24. |
| 0008 8038h | IWDT | IWDT Count Stop Control Register | IWDTCSTPR | 8 | 8 | 2 or 3 PCLKB | section 24. |
| 0008 80C0h | DA | D/A Data Register 0 | DADR0 | 16 | 16 | 2 or 3 PCLKB | section 31. |
| 0008 80C2h | DA | D/A Data Register 1 | DADR1 | 16 | 16 | 2 or 3 PCLKB | section 31. |
| 0008 80C4h | DA | D/A Control Register | DACR | 8 | 8 | 2 or 3 PCLKB | section 31. |
| 0008 80C5h | DA | DADRm Format Select Register | DADPR | 8 | 8 | 2 or 3 PCLKB | section 31. |
| 0008 8280h | CRC | CRC Control Register | CRCCR | 8 | 8 | 2 or 3 PCLKB | section 29. |
| 0008 8281h | CRC | CRC Data Input Register | CRCDIR | 8 | 8 | 2 or 3 PCLKB | section 29. |
| 0008 8282h | CRC | CRC Data Output Register | CRCDOR | 16 | 16 | 2 or 3 PCLKB | section 29. |
| 0008 8300h | RIIC0 | $I^2C$ Bus Control Register 1 | ICCR1 | 8 | 8 | 2 or 3 PCLKB | section 27. |
| 0008 8301h | RIIC0 | $I^2C$ Bus Control Register 2 | ICCR2 | 8 | 8 | 2 or 3 PCLKB | section 27. |
| 0008 8302h | RIIC0 | $I^2C$ Bus Mode Register 1 | ICMR1 | 8 | 8 | 2 or 3 PCLKB | section 27. |
| 0008 8303h | RIIC0 | $I^2C$ Bus Mode Register 2 | ICMR2 | 8 | 8 | 2 or 3 PCLKB | section 27. |
| 0008 8304h | RIIC0 | $I^2C$ Bus Mode Register 3 | ICMR3 | 8 | 8 | 2 or 3 PCLKB | section 27. |
| 0008 8305h | RIIC0 | $I^2C$ Bus Function Enable Register | ICFER | 8 | 8 | 2 or 3 PCLKB | section 27. |
| 0008 8306h | RIIC0 | $I^2C$ Bus Status Enable Register | ICSER | 8 | 8 | 2 or 3 PCLKB | section 27. |
| 0008 8307h | RIIC0 | $I^2C$ Bus Interrupt Enable Register | ICIER | 8 | 8 | 2 or 3 PCLKB | section 27. |
| 0008 8308h | RIIC0 | $I^2C$ Bus Status Register 1 | ICSR1 | 8 | 8 | 2 or 3 PCLKB | section 27. |
| 0008 8309h | RIIC0 | $I^2C$ Bus Status Register 2 | ICSR2 | 8 | 8 | 2 or 3 PCLKB | section 27. |
| 0008 830Ah | RIIC0 | Slave Address Register L0 | SARL0 | 8 | 8 | 2 or 3 PCLKB | section 27. |
| 0008 830Ah | RIIC0 | Timeout Internal Counter L | TMOCNTL | 8 | 8 | 2 or 3 PCLKB | section 27. |
| 0008 830Bh | RIIC0 | Slave Address Register U0 | SARU0 | 8 | 8 | 2 or 3 PCLKB | section 27. |
| 0008 830Bh | RIIC0 | Timeout Internal Counter U | TMOCNTU | 8 | 8 *1 | 2 or 3 PCLKB | section 27. |
| 0008 830Ch | RIIC0 | Slave Address Register L1 | SARL1 | 8 | 8 | 2 or 3 PCLKB | section 27. |
| 0008 830Dh | RIIC0 | Slave Address Register U1 | SARU1 | 8 | 8 | 2 or 3 PCLKB | section 27. |
| 0008 830Eh | RIIC0 | Slave Address Register L2 | SARL2 | 8 | 8 | 2 or 3 PCLKB | section 27. |
| 0008 830Fh | RIIC0 | Slave Address Register U2 | SARU2 | 8 | 8 | 2 or 3 PCLKB | section 27. |
| 0008 8310h | RIIC0 | $I^2C$ Bus Bit Rate Low-Level Register | ICBRL | 8 | 8 | 2 or 3 PCLKB | section 27. |
| 0008 8311h | RIIC0 | $I^2C$ Bus Bit Rate High-Level Register | ICBRH | 8 | 8 | 2 or 3 PCLKB | section 27. |
| 0008 8312h | RIIC0 | $I^2C$ Bus Transmit Data Register | ICDRT | 8 | 8 | 2 or 3 PCLKB | section 27. |
| 0008 8313h | RIIC0 | $I^2C$ Bus Receive Data Register | ICDRR | 8 | 8 | 2 or 3 PCLKB | section 27. |
| 0008 8380h | RSPI0 | RSPI Control Register | SPCR | 8 | 8 | 2 or 3 PCLKB | section 28. |
| 0008 8381h | RSPI0 | RSPI Slave Select Polarity Register | SSLP | 8 | 8 | 2 or 3 PCLKB | section 28. |
| 0008 8382h | RSPI0 | RSPI Pin Control Register | SPPCR | 8 | 8 | 2 or 3 PCLKB | section 28. |
| 0008 8383h | RSPI0 | RSPI Status Register | SPSR | 8 | 8 | 2 or 3 PCLKB | section 28. |
| 0008 8384h | RSPI0 | RSPI Data Register | SPDR | 32 | 16, 32 | 2 or 3 PCLKB/2ICLK | section 28. |
| 0008 8388h | RSPI0 | RSPI Sequence Control Register | SPSCR | 8 | 8 | 2 or 3 PCLKB | section 28. |
| 0008 8389h | RSPI0 | RSPI Sequence Status Register | SPSSR | 8 | 8 | 2 or 3 PCLKB | section 28. |
| 0008 838Ah | RSPI0 | RSPI Bit Rate Register | SPBR | 8 | 8 | 2 or 3 PCLKB | section 28. |
| 0008 838Bh | RSPI0 | RSPI Data Control Register | SPDCR | 8 | 8 | 2 or 3 PCLKB | section 28. |

**Table 5.1 List of I/O Registers (Address Order) (7/16)**

| Address | Module Symbol | Register Name | Register Symbol | Number of Bits | Access Size | Number of Access States | Reference Section |
|---|---|---|---|---|---|---|---|
| 0008 838Ch | RSPI0 | RSPI Clock Delay Register | SPCKD | 8 | 8 | 2 or 3 PCLKB | section 28. |
| 0008 838Dh | RSPI0 | RSPI Slave Select Negation Delay Register | SSLND | 8 | 8 | 2 or 3 PCLKB | section 28. |
| 0008 838Eh | RSPI0 | RSPI Next-Access Delay Register | SPND | 8 | 8 | 2 or 3 PCLKB | section 28. |
| 0008 838Fh | RSPI0 | RSPI Control Register 2 | SPCR2 | 8 | 8 | 2 or 3 PCLKB | section 28. |
| 0008 8390h | RSPI0 | RSPI Command Register 0 | SPCMD0 | 16 | 16 | 2 or 3 PCLKB | section 28. |
| 0008 8392h | RSPI0 | RSPI Command Register 1 | SPCMD1 | 16 | 16 | 2 or 3 PCLKB | section 28. |
| 0008 8394h | RSPI0 | RSPI Command Register 2 | SPCMD2 | 16 | 16 | 2 or 3 PCLKB | section 28. |
| 0008 8396h | RSPI0 | RSPI Command Register 3 | SPCMD3 | 16 | 16 | 2 or 3 PCLKB | section 28. |
| 0008 8398h | RSPI0 | RSPI Command Register 4 | SPCMD4 | 16 | 16 | 2 or 3 PCLKB | section 28. |
| 0008 839Ah | RSPI0 | RSPI Command Register 5 | SPCMD5 | 16 | 16 | 2 or 3 PCLKB | section 28. |
| 0008 839Ch | RSPI0 | RSPI Command Register 6 | SPCMD6 | 16 | 16 | 2 or 3 PCLKB | section 28. |
| 0008 839Eh | RSPI0 | RSPI Command Register 7 | SPCMD7 | 16 | 16 | 2 or 3 PCLKB | section 28. |
| 0008 8600h | MTU3 | Timer Control Register | TCR | 8 | 8 | 2 or 3 PCLKB | section 20. |
| 0008 8601h | MTU4 | Timer Control Register | TCR | 8 | 8 | 2 or 3 PCLKB | section 20. |
| 0008 8602h | MTU3 | Timer Mode Register | TMDR | 8 | 8 | 2 or 3 PCLKB | section 20. |
| 0008 8603h | MTU4 | Timer Mode Register | TMDR | 8 | 8 | 2 or 3 PCLKB | section 20. |
| 0008 8604h | MTU3 | Timer I/O Control Register H | TIORH | 8 | 8 | 2 or 3 PCLKB | section 20. |
| 0008 8605h | MTU3 | Timer I/O Control Register L | TIORL | 8 | 8 | 2 or 3 PCLKB | section 20. |
| 0008 8606h | MTU4 | Timer I/O Control Register H | TIORH | 8 | 8 | 2 or 3 PCLKB | section 20. |
| 0008 8607h | MTU4 | Timer I/O Control Register L | TIORL | 8 | 8 | 2 or 3 PCLKB | section 20. |
| 0008 8608h | MTU3 | Timer Interrupt Enable Register | TIER | 8 | 8 | 2 or 3 PCLKB | section 20. |
| 0008 8609h | MTU4 | Timer Interrupt Enable Register | TIER | 8 | 8 | 2 or 3 PCLKB | section 20. |
| 0008 860Ah | MTU | Timer Output Master Enable Register | TOER | 8 | 8 | 2 or 3 PCLKB | section 20. |
| 0008 860Dh | MTU | Timer Gate Control Register | TGCR | 8 | 8 | 2 or 3 PCLKB | section 20. |
| 0008 860Eh | MTU | Timer Output Control Register 1 | TOCR1 | 8 | 8 | 2 or 3 PCLKB | section 20. |
| 0008 860Fh | MTU | Timer Output Control Register 2 | TOCR2 | 8 | 8 | 2 or 3 PCLKB | section 20. |
| 0008 8610h | MTU3 | Timer Counter | TCNT | 16 | 16 | 2 or 3 PCLKB | section 20. |
| 0008 8612h | MTU4 | Timer Counter | TCNT | 16 | 16 | 2 or 3 PCLKB | section 20. |
| 0008 8614h | MTU | Timer Cycle Data Register | TCDR | 16 | 16 | 2 or 3 PCLKB | section 20. |
| 0008 8616h | MTU | Timer Dead Time Data Register | TDDR | 16 | 16 | 2 or 3 PCLKB | section 20. |
| 0008 8618h | MTU3 | Timer General Register A | TGRA | 16 | 16 | 2 or 3 PCLKB | section 20. |
| 0008 861Ah | MTU3 | Timer General Register B | TGRB | 16 | 16 | 2 or 3 PCLKB | section 20. |
| 0008 861Ch | MTU4 | Timer General Register A | TGRA | 16 | 16 | 2 or 3 PCLKB | section 20. |
| 0008 861Eh | MTU4 | Timer General Register B | TGRB | 16 | 16 | 2 or 3 PCLKB | section 20. |
| 0008 8620h | MTU | Timer Subcounter | TCNTS | 16 | 16 | 2 or 3 PCLKB | section 20. |
| 0008 8622h | MTU | Timer Cycle Buffer Register | TCBR | 16 | 16 | 2 or 3 PCLKB | section 20. |
| 0008 8624h | MTU3 | Timer General Register C | TGRC | 16 | 16 | 2 or 3 PCLKB | section 20. |
| 0008 8626h | MTU3 | Timer General Register D | TGRD | 16 | 16 | 2 or 3 PCLKB | section 20. |
| 0008 8628h | MTU4 | Timer General Register C | TGRC | 16 | 16 | 2 or 3 PCLKB | section 20. |
| 0008 862Ah | MTU4 | Timer General Register D | TGRD | 16 | 16 | 2 or 3 PCLKB | section 20. |
| 0008 862Ch | MTU3 | Timer Status Register | TSR | 8 | 8 | 2 or 3 PCLKB | section 20. |
| 0008 862Dh | MTU4 | Timer Status Register | TSR | 8 | 8 | 2 or 3 PCLKB | section 20. |
| 0008 8630h | MTU | Timer Interrupt Skipping Set Register | TITCR | 8 | 8 | 2 or 3 PCLKB | section 20. |
| 0008 8631h | MTU | Timer Interrupt Skipping Counter | TITCNT | 8 | 8 | 2 or 3 PCLKB | section 20. |
| 0008 8632h | MTU | Timer Buffer Transfer Set Register | TBTER | 8 | 8 | 2 or 3 PCLKB | section 20. |
| 0008 8634h | MTU | Timer Dead Time Enable Register | TDER | 8 | 8 | 2 or 3 PCLKB | section 20. |
| 0008 8636h | MTU | Timer Output Level Buffer Register | TOLBR | 8 | 8 | 2 or 3 PCLKB | section 20. |
| 0008 8638h | MTU3 | Timer Buffer Operation Transfer Mode Register | TBTM | 8 | 8 | 2 or 3 PCLKB | section 20. |
| 0008 8639h | MTU4 | Timer Buffer Operation Transfer Mode Register | TBTM | 8 | 8 | 2 or 3 PCLKB | section 20. |
| 0008 8640h | MTU4 | Timer A/D Converter Start Request Control Register | TADCR | 16 | 16 | 2 or 3 PCLKB | section 20. |
| 0008 8644h | MTU4 | Timer A/D Converter Start Request Cycle Set Register A | TADCORA | 16 | 16 | 2 or 3 PCLKB | section 20. |
| 0008 8646h | MTU4 | Timer A/D Converter Start Request Cycle Set Register B | TADCORB | 16 | 16 | 2 or 3 PCLKB | section 20. |

**Table 5.1 List of I/O Registers (Address Order) (8/16)**

| Address | Module Symbol | Register Name | Register Symbol | Number of Bits | Access Size | Number of Access States | Reference Section |
|---|---|---|---|---|---|---|---|
| 0008 8648h | MTU4 | Timer A/D Converter Start Request Cycle Set Buffer Register A | TADCOBRA | 16 | 16 | 2 or 3 PCLKB | section 20. |
| 0008 864Ah | MTU4 | Timer A/D Converter Start Request Cycle Set Buffer Register B | TADCOBRB | 16 | 16 | 2 or 3 PCLKB | section 20. |
| 0008 8660h | MTU | Timer Waveform Control Register | TWCR | 8 | 8, 16 | 2 or 3 PCLKB | section 20. |
| 0008 8680h | MTU | Timer Start Register | TSTR | 8 | 8, 16 | 2 or 3 PCLKB | section 20. |
| 0008 8681h | MTU | Timer Synchronous Register | TSYR | 8 | 8, 16 | 2 or 3 PCLKB | section 20. |
| 0008 8684h | MTU | Timer Read/Write Enable Register | TRWER | 8 | 8, 16 | 2 or 3 PCLKB | section 20. |
| 0008 8690h | MTU0 | Noise Filter Control Register | NFCR | 8 | 8, 16 | 2 or 3 PCLKB | section 20. |
| 0008 8691h | MTU1 | Noise Filter Control Register | NFCR | 8 | 8, 16 | 2 or 3 PCLKB | section 20. |
| 0008 8692h | MTU2 | Noise Filter Control Register | NFCR | 8 | 8, 16 | 2 or 3 PCLKB | section 20. |
| 0008 8693h | MTU3 | Noise Filter Control Register | NFCR | 8 | 8, 16 | 2 or 3 PCLKB | section 20. |
| 0008 8694h | MTU4 | Noise Filter Control Register | NFCR | 8 | 8, 16 | 2 or 3 PCLKB | section 20. |
| 0008 8695h | MTU5 | Noise Filter Control Register | NFCR | 8 | 8, 16 | 2 or 3 PCLKB | section 20. |
| 0008 8700h | MTU0 | Timer Control Register | TCR | 8 | 8 | 2 or 3 PCLKB | section 20. |
| 0008 8701h | MTU0 | Timer Mode Register | TMDR | 8 | 8 | 2 or 3 PCLKB | section 20. |
| 0008 8702h | MTU0 | Timer I/O Control Register H | TIORH | 8 | 8 | 2 or 3 PCLKB | section 20. |
| 0008 8703h | MTU0 | Timer I/O Control Register L | TIORL | 8 | 8 | 2 or 3 PCLKB | section 20. |
| 0008 8704h | MTU0 | Timer Interrupt Enable Register | TIER | 8 | 8 | 2 or 3 PCLKB | section 20. |
| 0008 8705h | MTU0 | Timer Status Register | TSR | 8 | 8 | 2 or 3 PCLKB | section 20. |
| 0008 8706h | MTU0 | Timer Counter | TCNT | 16 | 16 | 2 or 3 PCLKB | section 20. |
| 0008 8708h | MTU0 | Timer General Register A | TGRA | 16 | 16 | 2 or 3 PCLKB | section 20. |
| 0008 870Ah | MTU0 | Timer General Register B | TGRB | 16 | 16 | 2 or 3 PCLKB | section 20. |
| 0008 870Ch | MTU0 | Timer General Register C | TGRC | 16 | 16 | 2 or 3 PCLKB | section 20. |
| 0008 870Eh | MTU0 | Timer General Register D | TGRD | 16 | 16 | 2 or 3 PCLKB | section 20. |
| 0008 8720h | MTU0 | Timer General Register E | TGRE | 16 | 16 | 2 or 3 PCLKB | section 20. |
| 0008 8722h | MTU0 | Timer General Register F | TGRF | 16 | 16 | 2 or 3 PCLKB | section 20. |
| 0008 8724h | MTU0 | Timer Interrupt Enable Register 2 | TIER2 | 8 | 8 | 2 or 3 PCLKB | section 20. |
| 0008 8726h | MTU0 | Timer Buffer Operation Transfer Mode Register | TBTM | 8 | 8 | 2 or 3 PCLKB | section 20. |
| 0008 8780h | MTU1 | Timer Control Register | TCR | 8 | 8 | 2 or 3 PCLKB | section 20. |
| 0008 8781h | MTU1 | Timer Mode Register | TMDR | 8 | 8 | 2 or 3 PCLKB | section 20. |
| 0008 8782h | MTU1 | Timer I/O Control Register | TIOR | 8 | 8 | 2 or 3 PCLKB | section 20. |
| 0008 8784h | MTU1 | Timer Interrupt Enable Register | TIER | 8 | 8 | 2 or 3 PCLKB | section 20. |
| 0008 8785h | MTU1 | Timer Status Register | TSR | 8 | 8 | 2 or 3 PCLKB | section 20. |
| 0008 8786h | MTU1 | Timer Counter | TCNT | 16 | 16 | 2 or 3 PCLKB | section 20. |
| 0008 8788h | MTU1 | Timer General Register A | TGRA | 16 | 16 | 2 or 3 PCLKB | section 20. |
| 0008 878Ah | MTU1 | Timer General Register B | TGRB | 16 | 16 | 2 or 3 PCLKB | section 20. |
| 0008 8790h | MTU1 | Timer Input Capture Control Register | TICCR | 8 | 8 | 2 or 3 PCLKB | section 20. |
| 0008 8800h | MTU2 | Timer Control Register | TCR | 8 | 8 | 2 or 3 PCLKB | section 20. |
| 0008 8801h | MTU2 | Timer Mode Register | TMDR | 8 | 8 | 2 or 3 PCLKB | section 20. |
| 0008 8802h | MTU2 | Timer I/O Control Register | TIOR | 8 | 8 | 2 or 3 PCLKB | section 20. |
| 0008 8804h | MTU2 | Timer Interrupt Enable Register | TIER | 8 | 8 | 2 or 3 PCLKB | section 20. |
| 0008 8805h | MTU2 | Timer Status Register | TSR | 8 | 8 | 2 or 3 PCLKB | section 20. |
| 0008 8806h | MTU2 | Timer Counter | TCNT | 16 | 16 | 2 or 3 PCLKB | section 20. |
| 0008 8808h | MTU2 | Timer General Register A | TGRA | 16 | 16 | 2 or 3 PCLKB | section 20. |
| 0008 880Ah | MTU2 | Timer General Register B | TGRB | 16 | 16 | 2 or 3 PCLKB | section 20. |
| 0008 8880h | MTU5 | Timer Counter U | TCNTU | 16 | 16 | 2 or 3 PCLKB | section 20. |
| 0008 8882h | MTU5 | Timer General Register U | TGRU | 16 | 16 | 2 or 3 PCLKB | section 20. |
| 0008 8884h | MTU5 | Timer Control Register U | TCRU | 8 | 8 | 2 or 3 PCLKB | section 20. |
| 0008 8886h | MTU5 | Timer I/O Control Register U | TIORU | 8 | 8 | 2 or 3 PCLKB | section 20. |
| 0008 8890h | MTU5 | Timer Counter V | TCNTV | 16 | 16 | 2 or 3 PCLKB | section 20. |
| 0008 8892h | MTU5 | Timer General Register V | TGRV | 16 | 16 | 2 or 3 PCLKB | section 20. |
| 0008 8894h | MTU5 | Timer Control Register V | TCRV | 8 | 8 | 2 or 3 PCLKB | section 20. |
| 0008 8896h | MTU5 | Timer I/O Control Register V | TIORV | 8 | 8 | 2 or 3 PCLKB | section 20. |

**Table 5.1 List of I/O Registers (Address Order) (9/16)**

| Address | Module Symbol | Register Name | Register Symbol | Number of Bits | Access Size | Number of Access States | Reference Section |
|---|---|---|---|---|---|---|---|
| 0008 88A0h | MTU5 | Timer Counter W | TCNTW | 16 | 16 | 2 or 3 PCLKB | section 20. |
| 0008 88A2h | MTU5 | Timer General Register W | TGRW | 16 | 16 | 2 or 3 PCLKB | section 20. |
| 0008 88A4h | MTU5 | Timer Control Register W | TCRW | 8 | 8 | 2 or 3 PCLKB | section 20. |
| 0008 88A6h | MTU5 | Timer I/O Control Register W | TIORW | 8 | 8 | 2 or 3 PCLKB | section 20. |
| 0008 88B2h | MTU5 | Timer Interrupt Enable Register | TIER | 8 | 8 | 2 or 3 PCLKB | section 20. |
| 0008 88B4h | MTU5 | Timer Start Register | TSTR | 8 | 8 | 2 or 3 PCLKB | section 20. |
| 0008 88B6h | MTU5 | Timer Compare Match Clear Register | TCNTCMPCLR | 8 | 8 | 2 or 3 PCLKB | section 20. |
| 0008 8900h | POE | Input Level Control/Status Register 1 | ICSR1 | 16 | 8, 16 | 2 or 3 PCLKB | section 21. |
| 0008 8902h | POE | Output Level Control/Status Register 1 | OCSR1 | 16 | 8, 16 | 2 or 3 PCLKB | section 21. |
| 0008 8908h | POE | Input Level Control/Status Register 2 | ICSR2 | 16 | 8, 16 | 2 or 3 PCLKB | section 21. |
| 0008 890Ah | POE | Software Port Output Enable Register | SPOER | 8 | 8 | 2 or 3 PCLKB | section 21. |
| 0008 890Bh | POE | Port Output Enable Control Register 1 | POECR1 | 8 | 8 | 2 or 3 PCLKB | section 21. |
| 0008 890Ch | POE | Port Output Enable Control Register 2 | POECR2 | 8 | 8 | 2 or 3 PCLKB | section 21. |
| 0008 890Eh | POE | Input Level Control/Status Register 3 | ICSR3 | 16 | 8, 16 | 2 or 3 PCLKB | section 21. |
| 0008 9000h | S12AD | A/D Control Register | ADCSR | 16 | 16 | 2 or 3 PCLKB | section 30. |
| 0008 9004h | S12AD | A/D Channel Select Register A | ADANSA | 16 | 16 | 2 or 3 PCLKB | section 30. |
| 0008 9008h | S12AD | A/D-Converted Value Addition Mode Select Register | ADADS | 16 | 16 | 2 or 3 PCLKB | section 30. |
| 0008 900Ch | S12AD | A/D-Converted Value Addition Count Select Register | ADADC | 8 | 8 | 2 or 3 PCLKB | section 30. |
| 0008 900Eh | S12AD | A/D Control Extended Register | ADCER | 16 | 16 | 2 or 3 PCLKB | section 30. |
| 0008 9010h | S12AD | A/D Start Trigger Select Register | ADSTRGR | 16 | 16 | 2 or 3 PCLKB | section 30. |
| 0008 9012h | S12AD | A/D Converted Extended Input Control Register | ADEXICR | 16 | 16 | 2 or 3 PCLKB | section 30. |
| 0008 9014h | S12AD | A/D Channel Select Register B | ADANSB | 16 | 16 | 2 or 3 PCLKB | section 30. |
| 0008 9018h | S12AD | A/D Data Duplication Register | ADDBLDR | 16 | 16 | 2 or 3 PCLKB | section 30. |
| 0008 901Ah | S12AD | A/D Temperature Sensor Data Register | ADTSDR | 16 | 16 | 2 or 3 PCLKB | section 30. |
| 0008 901Ch | S12AD | A/D Internal Reference Voltage Data Register | ADOCDR | 16 | 16 | 2 or 3 PCLKB | section 30. |
| 0008 9020h | S12AD | A/D Data Register 0 | ADDR0 | 16 | 16 | 2 or 3 PCLKB | section 30. |
| 0008 9022h | S12AD | A/D Data Register 1 | ADDR1 | 16 | 16 | 2 or 3 PCLKB | section 30. |
| 0008 9024h | S12AD | A/D Data Register 2 | ADDR2 | 16 | 16 | 2 or 3 PCLKB | section 30. |
| 0008 9026h | S12AD | A/D Data Register 3 | ADDR3 | 16 | 16 | 2 or 3 PCLKB | section 30. |
| 0008 9028h | S12AD | A/D Data Register 4 | ADDR4 | 16 | 16 | 2 or 3 PCLKB | section 30. |
| 0008 902Ch | S12AD | A/D Data Register 6 | ADDR6 | 16 | 16 | 2 or 3 PCLKB | section 30. |
| 0008 9030h | S12AD | A/D Data Register 8 | ADDR8 | 16 | 16 | 2 or 3 PCLKB | section 30. |
| 0008 9032h | S12AD | A/D Data Register 9 | ADDR9 | 16 | 16 | 2 or 3 PCLKB | section 30. |
| 0008 9034h | S12AD | A/D Data Register 10 | ADDR10 | 16 | 16 | 2 or 3 PCLKB | section 30. |
| 0008 9036h | S12AD | A/D Data Register 11 | ADDR11 | 16 | 16 | 2 or 3 PCLKB | section 30. |
| 0008 9038h | S12AD | A/D Data Register 12 | ADDR12 | 16 | 16 | 2 or 3 PCLKB | section 30. |
| 0008 903Ah | S12AD | A/D Data Register 13 | ADDR13 | 16 | 16 | 2 or 3 PCLKB | section 30. |
| 0008 903Ch | S12AD | A/D Data Register 14 | ADDR14 | 16 | 16 | 2 or 3 PCLKB | section 30. |
| 0008 903Eh | S12AD | A/D Data Register 15 | ADDR15 | 16 | 16 | 2 or 3 PCLKB | section 30. |
| 0008 9060h | S12AD | A/D Sampling State Register 0 | ADSSTR0 | 8 | 8 | 2 or 3 PCLKB | section 30. |
| 0008 9061h | S12AD | A/D Sampling State Register L | ADSSTRL | 8 | 8 | 2 or 3 PCLKB | section 30. |
| 0008 9070h | S12AD | A/D Sampling State Register T | ADSSTRT | 8 | 8 | 2 or 3 PCLKB | section 30. |
| 0008 9071h | S12AD | A/D Sampling State Register O | ADSSTRO | 8 | 8 | 2 or 3 PCLKB | section 30. |
| 0008 9073h | S12AD | A/D Sampling State Register 1 | ADSSTR1 | 8 | 8 | 2 or 3 PCLKB | section 30. |
| 0008 9074h | S12AD | A/D Sampling State Register 2 | ADSSTR2 | 8 | 8 | 2 or 3 PCLKB | section 30. |
| 0008 9075h | S12AD | A/D Sampling State Register 3 | ADSSTR3 | 8 | 8 | 2 or 3 PCLKB | section 30. |
| 0008 9076h | S12AD | A/D Sampling State Register 4 | ADSSTR4 | 8 | 8 | 2 or 3 PCLKB | section 30. |
| 0008 9078h | S12AD | A/D Sampling State Register 6 | ADSSTR6 | 8 | 8 | 2 or 3 PCLKB | section 30. |
| 0008 A020h | SCI1 | Serial Mode Register | SMR | 8 | 8 | 2 or 3 PCLKB | section 26. |
| 0008 A021h | SCI1 | Bit Rate Register | BRR | 8 | 8 | 2 or 3 PCLKB | section 26. |
| 0008 A022h | SCI1 | Serial Control Register | SCR | 8 | 8 | 2 or 3 PCLKB | section 26. |
| 0008 A023h | SCI1 | Transmit Data Register | TDR | 8 | 8 | 2 or 3 PCLKB | section 26. |

**Table 5.1 List of I/O Registers (Address Order) (10/16)**

| Address | Module Symbol | Register Name | Register Symbol | Number of Bits | Access Size | Number of Access States | Reference Section |
|---|---|---|---|---|---|---|---|
| 0008 A024h | SCI1 | Serial Status Register | SSR | 8 | 8 | 2 or 3 PCLKB | section 26. |
| 0008 A025h | SCI1 | Receive Data Register | RDR | 8 | 8 | 2 or 3 PCLKB | section 26. |
| 0008 A026h | SCI1 | Smart Card Mode Register | SCMR | 8 | 8 | 2 or 3 PCLKB | section 26. |
| 0008 A027h | SCI1 | Serial Extended Mode Register | SEMR | 8 | 8 | 2 or 3 PCLKB | section 26. |
| 0008 A028h | SCI1 | Noise Filter Setting Register | SNFR | 8 | 8 | 2 or 3 PCLKB | section 26. |
| 0008 A029h | SCI1 | $I^2C$ Mode Register 1 | SIMR1 | 8 | 8 | 2 or 3 PCLKB | section 26. |
| 0008 A02Ah | SCI1 | $I^2C$ Mode Register 2 | SIMR2 | 8 | 8 | 2 or 3 PCLKB | section 26. |
| 0008 A02Bh | SCI1 | $I^2C$ Mode Register 3 | SIMR3 | 8 | 8 | 2 or 3 PCLKB | section 26. |
| 0008 A02Ch | SCI1 | $I^2C$ Status Register | SISR | 8 | 8 | 2 or 3 PCLKB | section 26. |
| 0008 A02Dh | SCI1 | SPI Mode Register | SPMR | 8 | 8 | 2 or 3 PCLKB | section 26. |
| 0008 A0A0h | SCI5 | Serial Mode Register | SMR | 8 | 8 | 2 or 3 PCLKB | section 26. |
| 0008 A0A1h | SCI5 | Bit Rate Register | BRR | 8 | 8 | 2 or 3 PCLKB | section 26. |
| 0008 A0A2h | SCI5 | Serial Control Register | SCR | 8 | 8 | 2 or 3 PCLKB | section 26. |
| 0008 A0A3h | SCI5 | Transmit Data Register | TDR | 8 | 8 | 2 or 3 PCLKB | section 26. |
| 0008 A0A4h | SCI5 | Serial Status Register | SSR | 8 | 8 | 2 or 3 PCLKB | section 26. |
| 0008 A0A5h | SCI5 | Receive Data Register | RDR | 8 | 8 | 2 or 3 PCLKB | section 26. |
| 0008 A0A6h | SCI5 | Smart Card Mode Register | SCMR | 8 | 8 | 2 or 3 PCLKB | section 26. |
| 0008 A0A7h | SCI5 | Serial Extended Mode Register | SEMR | 8 | 8 | 2 or 3 PCLKB | section 26. |
| 0008 A0A8h | SCI5 | Noise Filter Setting Register | SNFR | 8 | 8 | 2 or 3 PCLKB | section 26. |
| 0008 A0A9h | SCI5 | $I^2C$ Mode Register 1 | SIMR1 | 8 | 8 | 2 or 3 PCLKB | section 26. |
| 0008 A0AAh | SCI5 | $I^2C$ Mode Register 2 | SIMR2 | 8 | 8 | 2 or 3 PCLKB | section 26. |
| 0008 A0ABh | SCI5 | $I^2C$ Mode Register 3 | SIMR3 | 8 | 8 | 2 or 3 PCLKB | section 26. |
| 0008 A0ACh | SCI5 | $I^2C$ Status Register | SISR | 8 | 8 | 2 or 3 PCLKB | section 26. |
| 0008 A0ADh | SCI5 | SPI Mode Register | SPMR | 8 | 8 | 2 or 3 PCLKB | section 26. |
| 0008 B000h | CAC | CAC Control Register 0 | CACR0 | 8 | 8 | 2 or 3 PCLKB | section 10. |
| 0008 B001h | CAC | CAC Control Register 1 | CACR1 | 8 | 8 | 2 or 3 PCLKB | section 10. |
| 0008 B002h | CAC | CAC Control Register 2 | CACR2 | 8 | 8 | 2 or 3 PCLKB | section 10. |
| 0008 B003h | CAC | CAC Interrupt Control Register | CAICR | 8 | 8 | 2 or 3 PCLKB | section 10. |
| 0008 B004h | CAC | CAC Status Register | CASTR | 8 | 8 | 2 or 3 PCLKB | section 10. |
| 0008 B006h | CAC | CAC Upper-Limit Value Setting Register | CAULVR | 16 | 16 | 2 or 3 PCLKB | section 10. |
| 0008 B008h | CAC | CAC Lower-Limit Value Setting Register | CALLVR | 16 | 16 | 2 or 3 PCLKB | section 10. |
| 0008 B00Ah | CAC | CAC Counter Buffer Register | CACNTBR | 16 | 16 | 2 or 3 PCLKB | section 10. |
| 0008 B080h | DOC | DOC Control Register | DOCR | 8 | 8 | 2 or 3 PCLKB | section 33. |
| 0008 B082h | DOC | DOC Data Input Register | DODIR | 16 | 16 | 2 or 3 PCLKB | section 33. |
| 0008 B084h | DOC | DOC Data Setting Register | DODSR | 16 | 16 | 2 or 3 PCLKB | section 33. |
| 0008 B100h | ELC | Event Link Control Register | ELCR | 8 | 8 | 2 or 3 PCLKB | section 17. |
| 0008 B102h | ELC | Event Link Setting Register 1 | ELSR1 | 8 | 8 | 2 or 3 PCLKB | section 17. |
| 0008 B103h | ELC | Event Link Setting Register 2 | ELSR2 | 8 | 8 | 2 or 3 PCLKB | section 17. |
| 0008 B104h | ELC | Event Link Setting Register 3 | ELSR3 | 8 | 8 | 2 or 3 PCLKB | section 17. |
| 0008 B105h | ELC | Event Link Setting Register 4 | ELSR4 | 8 | 8 | 2 or 3 PCLKB | section 17. |
| 0008 B108h | ELC | Event Link Setting Register 7 | ELSR7 | 8 | 8 | 2 or 3 PCLKB | section 17. |
| 0008 B110h | ELC | Event Link Setting Register 15 | ELSR15 | 8 | 8 | 2 or 3 PCLKB | section 17. |
| 0008 B111h | ELC | Event Link Setting Register 16 | ELSR16 | 8 | 8 | 2 or 3 PCLKB | section 17. |
| 0008 B113h | ELC | Event Link Setting Register 18 | ELSR18 | 8 | 8 | 2 or 3 PCLKB | section 17. |
| 0008 B115h | ELC | Event Link Setting Register 20 | ELSR20 | 8 | 8 | 2 or 3 PCLKB | section 17. |
| 0008 B117h | ELC | Event Link Setting Register 22 | ELSR22 | 8 | 8 | 2 or 3 PCLKB | section 17. |
| 0008 B119h | ELC | Event Link Setting Register 24 | ELSR24 | 8 | 8 | 2 or 3 PCLKB | section 17. |
| 0008 B11Ah | ELC | Event Link Setting Register 25 | ELSR25 | 8 | 8 | 2 or 3 PCLKB | section 17. |
| 0008 B11Fh | ELC | Event Link Option Setting Register A | ELOPA | 8 | 8 | 2 or 3 PCLKB | section 17. |
| 0008 B120h | ELC | Event Link Option Setting Register B | ELOPB | 8 | 8 | 2 or 3 PCLKB | section 17. |
| 0008 B121h | ELC | Event Link Option Setting Register C | ELOPC | 8 | 8 | 2 or 3 PCLKB | section 17. |
| 0008 B123h | ELC | Port Group Setting Register 1 | PGR1 | 8 | 8 | 2 or 3 PCLKB | section 17. |

**Table 5.1 List of I/O Registers (Address Order) (11/16)**

| Address | Module Symbol | Register Name | Register Symbol | Number of Bits | Access Size | Number of Access States | Reference Section |
|---|---|---|---|---|---|---|---|
| 0008 B125h | ELC | Port Group Control Register 1 | PGC1 | 8 | 8 | 2 or 3 PCLKB | section 17. |
| 0008 B127h | ELC | Port Buffer Register 1 | PDBF1 | 8 | 8 | 2 or 3 PCLKB | section 17. |
| 0008 B129h | ELC | Event Link Port Setting Register 0 | PEL0 | 8 | 8 | 2 or 3 PCLKB | section 17. |
| 0008 B12Ah | ELC | Event Link Port Setting Register 1 | PEL1 | 8 | 8 | 2 or 3 PCLKB | section 17. |
| 0008 B12Dh | ELC | Event Link Software Event Generation Register | ELSEGR | 8 | 8 | 2 or 3 PCLKB | section 17. |
| 0008 B300h | SCI12 | Serial Mode Register | SMR | 8 | 8 | 2 or 3 PCLKB | section 26. |
| 0008 B301h | SCI12 | Bit Rate Register | BRR | 8 | 8 | 2 or 3 PCLKB | section 26. |
| 0008 B302h | SCI12 | Serial Control Register | SCR | 8 | 8 | 2 or 3 PCLKB | section 26. |
| 0008 B303h | SCI12 | Transmit Data Register | TDR | 8 | 8 | 2 or 3 PCLKB | section 26. |
| 0008 B304h | SCI12 | Serial Status Register | SSR | 8 | 8 | 2 or 3 PCLKB | section 26. |
| 0008 B305h | SCI12 | Receive Data Register | RDR | 8 | 8 | 2 or 3 PCLKB | section 26. |
| 0008 B306h | SCI12 | Smart Card Mode Register | SCMR | 8 | 8 | 2 or 3 PCLKB | section 26. |
| 0008 B307h | SCI12 | Serial Extended Mode Register | SEMR | 8 | 8 | 2 or 3 PCLKB | section 26. |
| 0008 B308h | SCI12 | Noise Filter Setting Register | SNFR | 8 | 8 | 2 or 3 PCLKB | section 26. |
| 0008 B309h | SCI12 | $I^2C$ Mode Register 1 | SIMR1 | 8 | 8 | 2 or 3 PCLKB | section 26. |
| 0008 B30Ah | SCI12 | $I^2C$ Mode Register 2 | SIMR2 | 8 | 8 | 2 or 3 PCLKB | section 26. |
| 0008 B30Bh | SCI12 | $I^2C$ Mode Register 3 | SIMR3 | 8 | 8 | 2 or 3 PCLKB | section 26. |
| 0008 B30Ch | SCI12 | $I^2C$ Status Register | SISR | 8 | 8 | 2 or 3 PCLKB | section 26. |
| 0008 B30Dh | SCI12 | SPI Mode Register | SPMR | 8 | 8 | 2 or 3 PCLKB | section 26. |
| 0008 B320h | SCI12 | Extended Serial Mode Enable Register | ESMER | 8 | 8 | 2 or 3 PCLKB | section 26. |
| 0008 B321h | SCI12 | Control Register 0 | CR0 | 8 | 8 | 2 or 3 PCLKB | section 26. |
| 0008 B322h | SCI12 | Control Register 1 | CR1 | 8 | 8 | 2 or 3 PCLKB | section 26. |
| 0008 B323h | SCI12 | Control Register 2 | CR2 | 8 | 8 | 2 or 3 PCLKB | section 26. |
| 0008 B324h | SCI12 | Control Register 3 | CR3 | 8 | 8 | 2 or 3 PCLKB | section 26. |
| 0008 B325h | SCI12 | Port Control Register | PCR | 8 | 8 | 2 or 3 PCLKB | section 26. |
| 0008 B326h | SCI12 | Interrupt Control Register | ICR | 8 | 8 | 2 or 3 PCLKB | section 26. |
| 0008 B327h | SCI12 | Status Register | STR | 8 | 8 | 2 or 3 PCLKB | section 26. |
| 0008 B328h | SCI12 | Status Clear Register | STCR | 8 | 8 | 2 or 3 PCLKB | section 26. |
| 0008 B329h | SCI12 | Control Field 0 Data Register | CF0DR | 8 | 8 | 2 or 3 PCLKB | section 26. |
| 0008 B32Ah | SCI12 | Control Field 0 Compare Enable Register | CF0CR | 8 | 8 | 2 or 3 PCLKB | section 26. |
| 0008 B32Bh | SCI12 | Control Field 0 Receive Data Register | CF0RR | 8 | 8 | 2 or 3 PCLKB | section 26. |
| 0008 B32Ch | SCI12 | Primary Control Field 1 Data Register | PCF1DR | 8 | 8 | 2 or 3 PCLKB | section 26. |
| 0008 B32Dh | SCI12 | Secondary Control Field 1 Data Register | SCF1DR | 8 | 8 | 2 or 3 PCLKB | section 26. |
| 0008 B32Eh | SCI12 | Control Field 1 Compare Enable Register | CF1CR | 8 | 8 | 2 or 3 PCLKB | section 26. |
| 0008 B32Fh | SCI12 | Control Field 1 Receive Data Register | CF1RR | 8 | 8 | 2 or 3 PCLKB | section 26. |
| 0008 B330h | SCI12 | Timer Control Register | TCR | 8 | 8 | 2 or 3 PCLKB | section 26. |
| 0008 B331h | SCI12 | Timer Mode Register | TMR | 8 | 8 | 2 or 3 PCLKB | section 26. |
| 0008 B332h | SCI12 | Timer Prescaler Register | TPRE | 8 | 8 | 2 or 3 PCLKB | section 26. |
| 0008 B333h | SCI12 | Timer Count Register | TCNT | 8 | 8 | 2 or 3 PCLKB | section 26. |
| 0008 C000h | PORT0 | Port Direction Register | PDR | 8 | 8 | 2 or 3 PCLKB | section 18. |
| 0008 C001h | PORT1 | Port Direction Register | PDR | 8 | 8 | 2 or 3 PCLKB | section 18. |
| 0008 C002h | PORT2 | Port Direction Register | PDR | 8 | 8 | 2 or 3 PCLKB | section 18. |
| 0008 C003h | PORT3 | Port Direction Register | PDR | 8 | 8 | 2 or 3 PCLKB | section 18. |
| 0008 C004h | PORT4 | Port Direction Register | PDR | 8 | 8 | 2 or 3 PCLKB | section 18. |
| 0008 C005h | PORT5 | Port Direction Register | PDR | 8 | 8 | 2 or 3 PCLKB | section 18. |
| 0008 C00Ah | PORTA | Port Direction Register | PDR | 8 | 8 | 2 or 3 PCLKB | section 18. |
| 0008 C00Bh | PORTB | Port Direction Register | PDR | 8 | 8 | 2 or 3 PCLKB | section 18. |
| 0008 C00Ch | PORTC | Port Direction Register | PDR | 8 | 8 | 2 or 3 PCLKB | section 18. |
| 0008 C00Eh | PORTE | Port Direction Register | PDR | 8 | 8 | 2 or 3 PCLKB | section 18. |
| 0008 C012h | PORTJ | Port Direction Register | PDR | 8 | 8 | 2 or 3 PCLKB | section 18. |
| 0008 C020h | PORT0 | Port Output Data Register | PODR | 8 | 8 | 2 or 3 PCLKB | section 18. |
| 0008 C021h | PORT1 | Port Output Data Register | PODR | 8 | 8 | 2 or 3 PCLKB | section 18. |

**Table 5.1 List of I/O Registers (Address Order) (12/16)**

| Address | Module Symbol | Register Name | Register Symbol | Number of Bits | Access Size | Number of Access States | Reference Section |
|---|---|---|---|---|---|---|---|
| 0008 C022h | PORT2 | Port Output Data Register | PODR | 8 | 8 | 2 or 3 PCLKB | section 18. |
| 0008 C023h | PORT3 | Port Output Data Register | PODR | 8 | 8 | 2 or 3 PCLKB | section 18. |
| 0008 C024h | PORT4 | Port Output Data Register | PODR | 8 | 8 | 2 or 3 PCLKB | section 18. |
| 0008 C025h | PORT5 | Port Output Data Register | PODR | 8 | 8 | 2 or 3 PCLKB | section 18. |
| 0008 C02Ah | PORTA | Port Output Data Register | PODR | 8 | 8 | 2 or 3 PCLKB | section 18. |
| 0008 C02Bh | PORTB | Port Output Data Register | PODR | 8 | 8 | 2 or 3 PCLKB | section 18. |
| 0008 C02Ch | PORTC | Port Output Data Register | PODR | 8 | 8 | 2 or 3 PCLKB | section 18. |
| 0008 C02Eh | PORTE | Port Output Data Register | PODR | 8 | 8 | 2 or 3 PCLKB | section 18. |
| 0008 C032h | PORTJ | Port Output Data Register | PODR | 8 | 8 | 2 or 3 PCLKB | section 18. |
| 0008 C040h | PORT0 | Port Input Data Register | PIDR | 8 | 8 | 3 or 4 PCLKB cycles when reading, 2 or 3 PCLKB cycles when writing | section 18. |
| 0008 C041h | PORT1 | Port Input Data Register | PIDR | 8 | 8 | 3 or 4 PCLKB cycles when reading, 2 or 3 PCLKB cycles when writing | section 18. |
| 0008 C042h | PORT2 | Port Input Data Register | PIDR | 8 | 8 | 3 or 4 PCLKB cycles when reading, 2 or 3 PCLKB cycles when writing | section 18. |
| 0008 C043h | PORT3 | Port Input Data Register | PIDR | 8 | 8 | 3 or 4 PCLKB cycles when reading, 2 or 3 PCLKB cycles when writing | section 18. |
| 0008 C044h | PORT4 | Port Input Data Register | PIDR | 8 | 8 | 3 or 4 PCLKB cycles when reading, 2 or 3 PCLKB cycles when writing | section 18. |
| 0008 C045h | PORT5 | Port Input Data Register | PIDR | 8 | 8 | 3 or 4 PCLKB cycles when reading, 2 or 3 PCLKB cycles when writing | section 18. |
| 0008 C04Ah | PORTA | Port Input Data Register | PIDR | 8 | 8 | 3 or 4 PCLKB cycles when reading, 2 or 3 PCLKB cycles when writing | section 18. |
| 0008 C04Bh | PORTB | Port Input Data Register | PIDR | 8 | 8 | 3 or 4 PCLKB cycles when reading, 2 or 3 PCLKB cycles when writing | section 18. |
| 0008 C04Ch | PORTC | Port Input Data Register | PIDR | 8 | 8 | 3 or 4 PCLKB cycles when reading, 2 or 3 PCLKB cycles when writing | section 18. |
| 0008 C04Eh | PORTE | Port Input Data Register | PIDR | 8 | 8 | 3 or 4 PCLKB cycles when reading, 2 or 3 PCLKB cycles when writing | section 18. |
| 0008 C051h | PORTH | Port Input Data Register | PIDR | 8 | 8 | 3 or 4 PCLKB cycles when reading, 2 or 3 PCLKB cycles when writing | section 18. |
| 0008 C052h | PORTJ | Port Input Data Register | PIDR | 8 | 8 | 3 or 4 PCLKB cycles when reading, 2 or 3 PCLKB cycles when writing | section 18. |
| 0008 C060h | PORT0 | Port Mode Register | PMR | 8 | 8 | 2 or 3 PCLKB | section 18. |
| 0008 C061h | PORT1 | Port Mode Register | PMR | 8 | 8 | 2 or 3 PCLKB | section 18. |
| 0008 C062h | PORT2 | Port Mode Register | PMR | 8 | 8 | 2 or 3 PCLKB | section 18. |
| 0008 C063h | PORT3 | Port Mode Register | PMR | 8 | 8 | 2 or 3 PCLKB | section 18. |
| 0008 C064h | PORT4 | Port Mode Register | PMR | 8 | 8 | 2 or 3 PCLKB | section 18. |
| 0008 C065h | PORT5 | Port Mode Register | PMR | 8 | 8 | 2 or 3 PCLKB | section 18. |
| 0008 C06Ah | PORTA | Port Mode Register | PMR | 8 | 8 | 2 or 3 PCLKB | section 18. |
| 0008 C06Bh | PORTB | Port Mode Register | PMR | 8 | 8 | 2 or 3 PCLKB | section 18. |
| 0008 C06Ch | PORTC | Port Mode Register | PMR | 8 | 8 | 2 or 3 PCLKB | section 18. |
| 0008 C06Eh | PORTE | Port Mode Register | PMR | 8 | 8 | 2 or 3 PCLKB | section 18. |
| 0008 C071h | PORTH | Port Mode Register | PMR | 8 | 8 | 2 or 3 PCLKB | section 18. |

**Table 5.1 List of I/O Registers (Address Order) (13/16)**

| Address | Module Symbol | Register Name | Register Symbol | Number of Bits | Access Size | Number of Access States | Reference Section |
|---|---|---|---|---|---|---|---|
| 0008 C072h | PORTJ | Port Mode Register | PMR | 8 | 8 | 2 or 3 PCLKB | section 18. |
| 0008 C083h | PORT1 | Open Drain Control Register 1 | ODR1 | 8 | 8, 16 | 2 or 3 PCLKB | section 18. |
| 0008 C085h | PORT2 | Open Drain Control Register 1 | ODR1 | 8 | 8, 16 | 2 or 3 PCLKB | section 18. |
| 0008 C086h | PORT3 | Open Drain Control Register 0 | ODR0 | 8 | 8, 16 | 2 or 3 PCLKB | section 18. |
| 0008 C094h | PORTA | Open Drain Control Register 0 | ODR0 | 8 | 8, 16 | 2 or 3 PCLKB | section 18. |
| 0008 C095h | PORTA | Open Drain Control Register 1 | ODR1 | 8 | 8, 16 | 2 or 3 PCLKB | section 18. |
| 0008 C096h | PORTB | Open Drain Control Register 0 | ODR0 | 8 | 8, 16 | 2 or 3 PCLKB | section 18. |
| 0008 C097h | PORTB | Open Drain Control Register 1 | ODR1 | 8 | 8, 16 | 2 or 3 PCLKB | section 18. |
| 0008 C098h | PORTC | Open Drain Control Register 0 | ODR0 | 8 | 8, 16 | 2 or 3 PCLKB | section 18. |
| 0008 C099h | PORTC | Open Drain Control Register 1 | ODR1 | 8 | 8, 16 | 2 or 3 PCLKB | section 18. |
| 0008 C09Ch | PORTE | Open Drain Control Register 0 | ODR0 | 8 | 8, 16 | 2 or 3 PCLKB | section 18. |
| 0008 C09Dh | PORTE | Open Drain Control Register 1 | ODR1 | 8 | 8, 16 | 2 or 3 PCLKB | section 18. |
| 0008 C0C0h | PORT0 | Pull-Up Control Register | PCR | 8 | 8 | 2 or 3 PCLKB | section 18. |
| 0008 C0C1h | PORT1 | Pull-Up Control Register | PCR | 8 | 8 | 2 or 3 PCLKB | section 18. |
| 0008 C0C2h | PORT2 | Pull-Up Control Register | PCR | 8 | 8 | 2 or 3 PCLKB | section 18. |
| 0008 C0C3h | PORT3 | Pull-Up Control Register | PCR | 8 | 8 | 2 or 3 PCLKB | section 18. |
| 0008 C0C5h | PORT5 | Pull-Up Control Register | PCR | 8 | 8 | 2 or 3 PCLKB | section 18. |
| 0008 C0CAh | PORTA | Pull-Up Control Register | PCR | 8 | 8 | 2 or 3 PCLKB | section 18. |
| 0008 C0CBh | PORTB | Pull-Up Control Register | PCR | 8 | 8 | 2 or 3 PCLKB | section 18. |
| 0008 C0CCh | PORTC | Pull-Up Control Register | PCR | 8 | 8 | 2 or 3 PCLKB | section 18. |
| 0008 C0CEh | PORTE | Pull-Up Control Register | PCR | 8 | 8 | 2 or 3 PCLKB | section 18. |
| 0008 C11Fh | MPC | Write-Protect Register | PWPR | 8 | 8 | 2 or 3 PCLKB | section 19. |
| 0008 C120h | PORT | Port Switching Register B | PSRB | 8 | 8 | 2 or 3 PCLKB | section 18. |
| 0008 C121h | PORT | Port Switching Register A | PSRA | 8 | 8 | 2 or 3 PCLKB | section 18. |
| 0008 C143h | MPC | P03 Pin Function Control Register | P03PFS | 8 | 8 | 2 or 3 PCLKB | section 19. |
| 0008 C145h | MPC | P05 Pin Function Control Register | P05PFS | 8 | 8 | 2 or 3 PCLKB | section 19. |
| 0008 C14Ch | MPC | P14 Pin Function Control Register | P14PFS | 8 | 8 | 2 or 3 PCLKB | section 19. |
| 0008 C14Dh | MPC | P15 Pin Function Control Register | P15PFS | 8 | 8 | 2 or 3 PCLKB | section 19. |
| 0008 C14Eh | MPC | P16 Pin Function Control Register | P16PFS | 8 | 8 | 2 or 3 PCLKB | section 19. |
| 0008 C14Fh | MPC | P17 Pin Function Control Register | P17PFS | 8 | 8 | 2 or 3 PCLKB | section 19. |
| 0008 C156h | MPC | P26 Pin Function Control Register | P26PFS | 8 | 8 | 2 or 3 PCLKB | section 19. |
| 0008 C157h | MPC | P27 Pin Function Control Register | P27PFS | 8 | 8 | 2 or 3 PCLKB | section 19. |
| 0008 C158h | MPC | P30 Pin Function Control Register | P30PFS | 8 | 8 | 2 or 3 PCLKB | section 19. |
| 0008 C159h | MPC | P31 Pin Function Control Register | P31PFS | 8 | 8 | 2 or 3 PCLKB | section 19. |
| 0008 C15Ah | MPC | P32 Pin Function Control Register | P32PFS | 8 | 8 | 2 or 3 PCLKB | section 19. |
| 0008 C15Dh | MPC | P35 Pin Function Control Register | P35PFS | 8 | 8 | 2 or 3 PCLKB | section 19. |
| 0008 C160h | MPC | P40 Pin Function Control Register | P40PFS | 8 | 8 | 2 or 3 PCLKB | section 19. |
| 0008 C161h | MPC | P41 Pin Function Control Register | P41PFS | 8 | 8 | 2 or 3 PCLKB | section 19. |
| 0008 C162h | MPC | P42 Pin Function Control Register | P42PFS | 8 | 8 | 2 or 3 PCLKB | section 19. |
| 0008 C163h | MPC | P43 Pin Function Control Register | P43PFS | 8 | 8 | 2 or 3 PCLKB | section 19. |
| 0008 C164h | MPC | P44 Pin Function Control Register | P44PFS | 8 | 8 | 2 or 3 PCLKB | section 19. |
| 0008 C166h | MPC | P46 Pin Function Control Register | P46PFS | 8 | 8 | 2 or 3 PCLKB | section 19. |
| 0008 C16Ch | MPC | P54 Pin Function Control Register | P54PFS | 8 | 8 | 2 or 3 PCLKB | section 19. |
| 0008 C16Dh | MPC | P55 Pin Function Control Register | P55PFS | 8 | 8 | 2 or 3 PCLKB | section 19. |
| 0008 C190h | MPC | PA0 Pin Function Control Register | PA0PFS | 8 | 8 | 2 or 3 PCLKB | section 19. |
| 0008 C191h | MPC | PA1 Pin Function Control Register | PA1PFS | 8 | 8 | 2 or 3 PCLKB | section 19. |
| 0008 C193h | MPC | PA3 Pin Function Control Register | PA3PFS | 8 | 8 | 2 or 3 PCLKB | section 19. |
| 0008 C194h | MPC | PA4 Pin Function Control Register | PA4PFS | 8 | 8 | 2 or 3 PCLKB | section 19. |
| 0008 C196h | MPC | PA6 Pin Function Control Register | PA6PFS | 8 | 8 | 2 or 3 PCLKB | section 19. |
| 0008 C198h | MPC | PB0 Pin Function Control Register | PB0PFS | 8 | 8 | 2 or 3 PCLKB | section 19. |
| 0008 C199h | MPC | PB1 Pin Function Control Register | PB1PFS | 8 | 8 | 2 or 3 PCLKB | section 19. |
| 0008 C19Bh | MPC | PB3 Pin Function Control Register | PB3PFS | 8 | 8 | 2 or 3 PCLKB | section 19. |

**Table 5.1 List of I/O Registers (Address Order) (14/16)**

| Address | Module Symbol | Register Name | Register Symbol | Number of Bits | Access Size | Number of Access States | Reference Section |
|---|---|---|---|---|---|---|---|
| 0008 C19Dh | MPC | PB5 Pin Function Control Register | PB5PFS | 8 | 8 | 2 or 3 PCLKB | section 19. |
| 0008 C19Eh | MPC | PB6 Pin Function Control Register | PB6PFS | 8 | 8 | 2 or 3 PCLKB | section 19. |
| 0008 C19Fh | MPC | PB7 Pin Function Control Register | PB7PFS | 8 | 8 | 2 or 3 PCLKB | section 19. |
| 0008 C1A2h | MPC | PC2 Pin Function Control Register | PC2PFS | 8 | 8 | 2 or 3 PCLKB | section 19. |
| 0008 C1A3h | MPC | PC3 Pin Function Control Register | PC3PFS | 8 | 8 | 2 or 3 PCLKB | section 19. |
| 0008 C1A4h | MPC | PC4 Pin Function Control Register | PC4PFS | 8 | 8 | 2 or 3 PCLKB | section 19. |
| 0008 C1A5h | MPC | PC5 Pin Function Control Register | PC5PFS | 8 | 8 | 2 or 3 PCLKB | section 19. |
| 0008 C1A6h | MPC | PC6 Pin Function Control Register | PC6PFS | 8 | 8 | 2 or 3 PCLKB | section 19. |
| 0008 C1A7h | MPC | PC7 Pin Function Control Register | PC7PFS | 8 | 8 | 2 or 3 PCLKB | section 19. |
| 0008 C1B0h | MPC | PE0 Pin Function Control Register | PE0PFS | 8 | 8 | 2 or 3 PCLKB | section 19. |
| 0008 C1B1h | MPC | PE1 Pin Function Control Register | PE1PFS | 8 | 8 | 2 or 3 PCLKB | section 19. |
| 0008 C1B2h | MPC | PE2 Pin Function Control Register | PE2PFS | 8 | 8 | 2 or 3 PCLKB | section 19. |
| 0008 C1B3h | MPC | PE3 Pin Function Control Register | PE3PFS | 8 | 8 | 2 or 3 PCLKB | section 19. |
| 0008 C1B4h | MPC | PE4 Pin Function Control Register | PE4PFS | 8 | 8 | 2 or 3 PCLKB | section 19. |
| 0008 C1B5h | MPC | PE5 Pin Function Control Register | PE5PFS | 8 | 8 | 2 or 3 PCLKB | section 19. |
| 0008 C1B6h | MPC | PE6 Pin Function Control Register | PE6PFS | 8 | 8 | 2 or 3 PCLKB | section 19. |
| 0008 C1B7h | MPC | PE7 Pin Function Control Register | PE7PFS | 8 | 8 | 2 or 3 PCLKB | section 19. |
| 0008 C1D6h | MPC | PJ6 Pin Function Control Register | PJ6PFS | 8 | 8 | 2 or 3 PCLKB | section 19. |
| 0008 C1D7h | MPC | PJ7 Pin Function Control Register | PJ7PFS | 8 | 8 | 2 or 3 PCLKB | section 19. |
| 0008 C290h | SYSTEM | Reset Status Register 0 | RSTSR0 | 8 | 8 | 4 or 5 PCLKB | section 6. |
| 0008 C291h | SYSTEM | Reset Status Register 1 | RSTSR1 | 8 | 8 | 4 or 5 PCLKB | section 6. |
| 0008 C293h | SYSTEM | Main Clock Oscillator Forced Oscillation Control Register | MOFCR | 8 | 8 | 4 or 5 PCLKB | section 9. |
| 0008 C297h | SYSTEM | Voltage Monitoring Circuit Control Register | LVCMPCR | 8 | 8 | 4 or 5 PCLKB | section 8. |
| 0008 C298h | SYSTEM | Voltage Detection Level Select Register | LVDLVLR | 8 | 8 | 4 or 5 PCLKB | section 8. |
| 0008 C29Ah | SYSTEM | Voltage Monitoring 1 Circuit Control Register 0 | LVD1CR0 | 8 | 8 | 4 or 5 PCLKB | section 8. |
| 0008 C29Bh | SYSTEM | Voltage Monitoring 2 Circuit Control Register 0 | LVD2CR0 | 8 | 8 | 4 or 5 PCLKB | section 8. |
| 0008 C400h | RTC | 64-Hz Counter | R64CNT | 8 | 8 | 2 or 3 PCLKB | section 23. |
| 0008 C402h | RTC | Second Counter | RSECCNT | 8 | 8 | 2 or 3 PCLKB | section 23. |
| 0008 C402h | RTC | Binary Counter 0 | BCNT0 | 8 | 8 | 2 or 3 PCLKB | section 23. |
| 0008 C404h | RTC | Minute Counter | RMINCNT | 8 | 8 | 2 or 3 PCLKB | section 23. |
| 0008 C404h | RTC | Binary Counter 1 | BCNT1 | 8 | 8 | 2 or 3 PCLKB | section 23. |
| 0008 C406h | RTC | Hour Counter | RHRCNT | 8 | 8 | 2 or 3 PCLKB | section 23. |
| 0008 C406h | RTC | Binary Counter 2 | BCNT2 | 8 | 8 | 2 or 3 PCLKB | section 23. |
| 0008 C408h | RTC | Day-Of-Week Counter | RWKCNT | 8 | 8 | 2 or 3 PCLKB | section 23. |
| 0008 C408h | RTC | Binary Counter 3 | BCNT3 | 8 | 8 | 2 or 3 PCLKB | section 23. |
| 0008 C40Ah | RTC | Date Counter | RDAYCNT | 8 | 8 | 2 or 3 PCLKB | section 23. |
| 0008 C40Ch | RTC | Month Counter | RMONCNT | 8 | 8 | 2 or 3 PCLKB | section 23. |
| 0008 C40Eh | RTC | Year Counter | RYRCNT | 16 | 16 | 2 or 3 PCLKB | section 23. |
| 0008 C410h | RTC | Second Alarm Register | RSECAR | 8 | 8 | 2 or 3 PCLKB | section 23. |
| 0008 C410h | RTC | Binary Counter 0 Alarm Register | BCNT0AR | 8 | 8 | 2 or 3 PCLKB | section 23. |
| 0008 C412h | RTC | Minute Alarm Register | RMINAR | 8 | 8 | 2 or 3 PCLKB | section 23. |
| 0008 C412h | RTC | Binary Counter 1 Alarm Register | BCNT1AR | 8 | 8 | 2 or 3 PCLKB | section 23. |
| 0008 C414h | RTC | Hour Alarm Register | RHRAR | 8 | 8 | 2 or 3 PCLKB | section 23. |
| 0008 C414h | RTC | Binary Counter 2 Alarm Register | BCNT2AR | 8 | 8 | 2 or 3 PCLKB | section 23. |
| 0008 C416h | RTC | Day-of-Week Alarm Register | RWKAR | 8 | 8 | 2 or 3 PCLKB | section 23. |
| 0008 C416h | RTC | Binary Counter 3 Alarm Register | BCNT3AR | 8 | 8 | 2 or 3 PCLKB | section 23. |
| 0008 C418h | RTC | Date Alarm Register | RDAYAR | 8 | 8 | 2 or 3 PCLKB | section 23. |
| 0008 C418h | RTC | Binary Counter 0 Alarm Enable Register | BCNT0AER | 8 | 8 | 2 or 3 PCLKB | section 23. |
| 0008 C41Ah | RTC | Month Alarm Register | RMONAR | 8 | 8 | 2 or 3 PCLKB | section 23. |
| 0008 C41Ah | RTC | Binary Counter 1 Alarm Enable Register | BCNT1AER | 8 | 8 | 2 or 3 PCLKB | section 23. |
| 0008 C41Ch | RTC | Year Alarm Register | RYRAR | 16 | 16 | 2 or 3 PCLKB | section 23. |
| 0008 C41Ch | RTC | Binary Counter 2 Alarm Enable Register | BCNT2AER | 16 | 16 | 2 or 3 PCLKB | section 23. |

**Table 5.1 List of I/O Registers (Address Order) (15/16)**

| Address | Module Symbol | Register Name | Register Symbol | Number of Bits | Access Size | Number of Access States | Reference Section |
|---|---|---|---|---|---|---|---|
| 0008 C41Eh | RTC | Year Alarm Enable Register | RYRAREN | 8 | 8 | 2 or 3 PCLKB | section 23. |
| 0008 C41Eh | RTC | Binary Counter 3 Alarm Enable Register | BCNT3AER | 8 | 8 | 2 or 3 PCLKB | section 23. |
| 0008 C422h | RTC | RTC Control Register 1 | RCR1 | 8 | 8 | 2 or 3 PCLKB | section 23. |
| 0008 C424h | RTC | RTC Control Register 2 | RCR2 | 8 | 8 | 2 or 3 PCLKB | section 23. |
| 0008 C426h | RTC | RTC Control Register 3 | RCR3 | 8 | 8 | 2 or 3 PCLKB | section 23. |
| 0008 C42Eh | RTC | Time Error Adjustment Register | RADJ | 8 | 8 | 2 or 3 PCLKB | section 23. |
| 000A 0000h | USB0 | System Configuration Control Register | SYSCFG | 16 | 16 | 3 or 4 PCLKB | section 25. |
| 000A 0004h | USB0 | System Configuration Status Register 0 | SYSSTS0 | 16 | 16 | 9 PCLK or more | section 25. |
| 000A 0008h | USB0 | Device State Control Register 0 | DVSTCTR0 | 16 | 16 | 9 PCLK or more | section 25. |
| 000A 0014h | USB0 | CFIFO Port Register | CFIFO | 16 | 16 | 3 or 4 PCLKB | section 25. |
| 000A 0018h | USB0 | D0FIFO Port Register | D0FIFO | 16 | 16 | 3 or 4 PCLKB | section 25. |
| 000A 001Ch | USB0 | D1FIFO Port Register | D1FIFO | 16 | 16 | 3 or 4 PCLKB | section 25. |
| 000A 0020h | USB0 | CFIFO Port Select Register | CFIFOSEL | 16 | 16 | 3 or 4 PCLKB | section 25. |
| 000A 0028h | USB0 | D0FIFO Port Select Register | D0FIFOSEL | 16 | 16 | 3 or 4 PCLKB | section 25. |
| 000A 002Ch | USB0 | D1FIFO Port Select Register | D1FIFOSEL | 16 | 16 | 3 or 4 PCLKB | section 25. |
| 000A 0022h | USB0 | CFIFO Port Control Register | CFIFOCTR | 16 | 16 | 3 or 4 PCLKB | section 25. |
| 000A 002Ah | USB0 | D0FIFO Port Control Register | D0FIFOCTR | 16 | 16 | 3 or 4 PCLKB | section 25. |
| 000A 002Eh | USB0 | D1FIFO Port Control Register | D1FIFOCTR | 16 | 16 | 3 or 4 PCLKB | section 25. |
| 000A 0030h | USB0 | Interrupt Enable Register 0 | INTENB0 | 16 | 16 | 9 PCLKB or more | section 25. |
| 000A 0032h | USB0 | Interrupt Enable Register 1 | INTENB1 | 16 | 16 | 9 PCLKB or more | section 25. |
| 000A 0036h | USB0 | BRDY Interrupt Enable Register | BRDYENB | 16 | 16 | 9 PCLKB or more | section 25. |
| 000A 0038h | USB0 | NRDY Interrupt Enable Register | NRDYENB | 16 | 16 | 9 PCLKB or more | section 25. |
| 000A 003Ah | USB0 | BEMP Interrupt Enable Register | BEMPENB | 16 | 16 | 9 PCLKB or more | section 25. |
| 000A 003Ch | USB0 | SOF Output Configuration Register | SOFCFG | 16 | 16 | 9 PCLKB or more | section 25. |
| 000A 0040h | USB0 | Interrupt Status Register 0 | INTSTS0 | 16 | 16 | 9 PCLKB or more | section 25. |
| 000A 0042h | USB0 | Interrupt Status Register 1 | INTSTS1 | 16 | 16 | 9 PCLKB or more | section 25. |
| 000A 0046h | USB0 | BRDY Interrupt Status Register | BRDYSTS | 16 | 16 | 9 PCLKB or more | section 25. |
| 000A 0048h | USB0 | NRDY Interrupt Status Register | NRDYSTS | 16 | 16 | 9 PCLKB or more | section 25. |
| 000A 004Ah | USB0 | BEMP Interrupt Status Register | BEMPSTS | 16 | 16 | 9 PCLKB or more | section 25. |
| 000A 004Ch | USB0 | Frame Number Register | FRMNUM | 16 | 16 | 9 PCLKB or more | section 25. |
| 000A 0054h | USB0 | USB Request Type Register | USBREQ | 16 | 16 | 9 PCLKB or more | section 25. |
| 000A 0056h | USB0 | USB Request Value Register | USBVAL | 16 | 16 | 9 PCLKB or more | section 25. |
| 000A 0058h | USB0 | USB Request Index Register | USBINDX | 16 | 16 | 9 PCLKB or more | section 25. |
| 000A 005Ah | USB0 | USB Request Length Register | USBLENG | 16 | 16 | 9 PCLKB or more | section 25. |
| 000A 005Ch | USB0 | DCP Configuration Register | DCPCFG | 16 | 16 | 9 PCLKB or more | section 25. |
| 000A 005Eh | USB0 | DCP Maximum Packet Size Register | DCPMAXP | 16 | 16 | 9 PCLKB or more | section 25. |
| 000A 0060h | USB0 | DCP Control Register | DCPCTR | 16 | 16 | 9 PCLKB or more | section 25. |
| 000A 0064h | USB0 | Pipe Window Select Register | PIPESEL | 16 | 16 | 9 PCLKB or more | section 25. |
| 000A 0068h | USB0 | Pipe Configuration Register | PIPECFG | 16 | 16 | 9 PCLKB or more | section 25. |
| 000A 006Ch | USB0 | Pipe Maximum Packet Size Register | PIPEMAXP | 16 | 16 | 9 PCLKB or more | section 25. |
| 000A 006Eh | USB0 | Pipe Cycle Control Register | PIPEPERI | 16 | 16 | 9 PCLKB or more | section 25. |
| 000A 0070h | USB0 | PIPE1 Control Register | PIPE1CTR | 16 | 16 | 9 PCLKB or more | section 25. |
| 000A 0072h | USB0 | PIPE2 Control Register | PIPE2CTR | 16 | 16 | 9 PCLKB or more | section 25. |
| 000A 0074h | USB0 | PIPE3 Control Register | PIPE3CTR | 16 | 16 | 9 PCLKB or more | section 25. |
| 000A 0076h | USB0 | PIPE4 Control Register | PIPE4CTR | 16 | 16 | 9 PCLKB or more | section 25. |
| 000A 0078h | USB0 | PIPE5 Control Register | PIPE5CTR | 16 | 16 | 9 PCLKB or more | section 25. |
| 000A 007Ah | USB0 | PIPE6 Control Register | PIPE6CTR | 16 | 16 | 9 PCLKB or more | section 25. |
| 000A 007Ch | USB0 | PIPE7 Control Register | PIPE7CTR | 16 | 16 | 9 PCLKB or more | section 25. |
| 000A 007Eh | USB0 | PIPE8 Control Register | PIPE8CTR | 16 | 16 | 9 PCLKB or more | section 25. |
| 000A 0080h | USB0 | PIPE9 Control Register | PIPE9CTR | 16 | 16 | 9 PCLKB or more | section 25. |
| 000A 0090h | USB0 | PIPE1 Transaction Counter Enable Register | PIPE1TRE | 16 | 16 | 9 PCLKB or more | section 25. |
| 000A 0092h | USB0 | PIPE1 Transaction Counter Register | PIPE1TRN | 16 | 16 | 9 PCLKB or more | section 25. |

**Table 5.1 List of I/O Registers (Address Order) (16/16)**

| Address | Module Symbol | Register Name | Register Symbol | Number of Bits | Access Size | Number of Access States | Reference Section |
|---|---|---|---|---|---|---|---|
| 000A 0094h | USB0 | PIPE2 Transaction Counter Enable Register | PIPE2TRE | 16 | 16 | 9 PCLKB or more | section 25. |
| 000A 0096h | USB0 | PIPE2 Transaction Counter Register | PIPE2TRN | 16 | 16 | 9 PCLKB or more | section 25. |
| 000A 0098h | USB0 | PIPE3 Transaction Counter Enable Register | PIPE3TRE | 16 | 16 | 9 PCLKB or more | section 25. |
| 000A 009Ah | USB0 | PIPE3 Transaction Counter Register | PIPE3TRN | 16 | 16 | 9 PCLKB or more | section 25. |
| 000A 009Ch | USB0 | PIPE4 Transaction Counter Enable Register | PIPE4TRE | 16 | 16 | 9 PCLKB or more | section 25. |
| 000A 009Eh | USB0 | PIPE4 Transaction Counter Register | PIPE4TRN | 16 | 16 | 9 PCLKB or more | section 25. |
| 000A 00A0h | USB0 | PIPE5 Transaction Counter Enable Register | PIPE5TRE | 16 | 16 | 9 PCLKB or more | section 25. |
| 000A 00A2h | USB0 | PIPE5 Transaction Counter Register | PIPE5TRN | 16 | 16 | 9 PCLKB or more | section 25. |
| 000A 00B0h | USB0 | BC Control Register 0 | USBBCCTRL0 | 16 | 16 | 9 PCLKB or more | section 25. |
| 000A 00CCh | USB0 | USB Module Control Register | USBMC | 16 | 16 | 9 PCLKB or more | section 25. |
| 000A 00D0h | USB0 | Device Address 0 Configuration Register | DEVADD0 | 16 | 16 | 9 PCLKB or more | section 25. |
| 000A 00D2h | USB0 | Device Address 1 Configuration Register | DEVADD1 | 16 | 16 | 9 PCLKB or more | section 25. |
| 000A 00D4h | USB0 | Device Address 2 Configuration Register | DEVADD2 | 16 | 16 | 9 PCLKB or more | section 25. |
| 000A 00D6h | USB0 | Device Address 3 Configuration Register | DEVADD3 | 16 | 16 | 9 PCLKB or more | section 25. |
| 000A 00D8h | USB0 | Device Address 4 Configuration Register | DEVADD4 | 16 | 16 | 9 PCLKB or more | section 25. |
| 000A 00DAh | USB0 | Device Address 5 Configuration Register | DEVADD5 | 16 | 16 | 9 PCLKB or more | section 25. |
| 007F C090h | FLASH | E2 DataFlash Control Register | DFLCTL | 8 | 8 | 2 or 3 FCLK | section 35. |

Note 1. Odd addresses should not be accessed in 16-bit units. When accessing a register in 16-bit units, access the address of the TMR0 or TMR2 register. Table 27.6 lists register allocation for 16-bit access.

# 6. Resets

## 6.1 Overview

There are six types of resets: RES# pin reset, power-on reset, voltage monitoring 1 reset, voltage monitoring 2 reset, independent watchdog timer reset, and software reset.
Table 6.1 lists the reset names and sources.

**Table 6.1 Reset Names and Sources**

| Reset Name | Source |
|---|---|
| RES# pin reset | Voltage input to the RES# pin is driven low. |
| Power-on reset | VCC rises (voltage monitored: VPOR)*1 |
| Voltage monitoring 1 reset | VCC falls (voltage monitored: Vdet1)*1 |
| Voltage monitoring 2 reset | VCC falls (voltage monitored: Vdet2)*1 |
| Independent watchdog timer reset | The independent watchdog timer underflows, or a refresh error occurs. |
| Software reset | Register setting |

Note 1. For details on the voltages to be monitored (VPOR, Vdet1, and Vdet2), refer to section 8, Voltage Detection Circuit (LVDAa) and section 36, Electrical Characteristics

The internal state and pins are initialized by a reset.
Table 6.2 lists the reset targets to be initialized.

**Table 6.2 Targets Initialized According to Reset Source**

| Target to be Initialized | Reset Source | | | | | |
|---|---|---|---|---|---|---|
| | RES# Pin Reset | Power-On Reset | Independent Watchdog Timer Reset | Voltage Monitoring 1 Reset | Voltage Monitoring 2 Reset | Software Reset |
| Power-on reset detect flag (RSTSR0.PORF) | ○ | — | — | — | — | — |
| Register related to the cold start/warm start determination flag (RSTSR1.CWSF) | — *1 | ○ | — | — | — | — |
| Independent watchdog timer reset detect flag (RSTSR2.IWDTRF) | ○ | ○ | — | — | — | — |
| Registers related to the independent watchdog timer (IWDTRR, IWDTCR, IWDTSR, IWDTRCR, IWDTCSTPR, ILOCOCR) | ○ | ○ | — | — | — | — |
| Voltage monitoring 1 reset detect flag (RSTSR0.LVD1RF) | ○ | ○ | ○ | — | — | — |
| Registers related to the voltage monitor function 1 (LVD1CR0, LVCMPCR.LVD1E, LVDLVLR.LVD1LVL[3:0]) | ○ | ○ | ○ | — | — | — |
| (LVD1CR1, LVD1SR) | ○ | ○ | ○ | — | — | — |
| Voltage monitoring 2 reset detect flag (RSTSR0.LVD2RF) | ○ | ○ | ○ | ○ | — | — |
| Registers related to the voltage monitor function 2 (LVD2CR0, EXVCCINP2, LVCMPCR.LVD2E, LVDLVLR.LVD2LVL[1:0]) | ○ | ○ | ○ | ○ | — | — |
| (LVD2CR1, LVD2SR) | ○ | ○ | ○ | ○ | — | — |
| Software reset detect flag (RSTSR2.SWRF) | ○ | ○ | ○ | ○ | ○ | — |
| Registers related to the realtime clock*2 | — | — | — | — | — | — |
| Registers other than the above, the CPU, and internal state | ○ | ○ | ○ | ○ | ○ | ○ |

○: Targets to be initialized, —: No change occurs.

Note 1. Initialized at a power-on.

Note 2. Some control bits (RCR1.CIE, RCR1.RTCOS, RCR2.RTCOE, ADJ30, and RESET) are initialized by all types of resets. For details on the target bits, refer to section 23, Realtime Clock (RTCA).

When a reset is canceled, the reset exception handling starts. For details on the reset exception handling, refer to section 13, Exception Handling.
Table 6.3 lists the pin related to the reset.

**Table 6.3 Pin Related to Reset**

| Pin Name | I/O | Function |
|---|---|---|
| RES# | Input | Reset pin |

## 6.2 Register Descriptions

### 6.2.1 Reset Status Register 0 (RSTSR0)

Address(es): 0008 C290h

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | LVD2RF | LVD1RF | — | PORF |
| Value after reset: | 0 | 0 | 0 | 0 | 0*1 | 0*1 | 0 | 0*1 |

Note 1. The value after reset depends on the reset source.

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b0 | PORF | Power-On Reset Detect Flag | 0: Power-on reset not detected.<br>1: Power-on reset detected. | R/(W) *1 |
| b1 | — | Reserved | This bit is read as 0. The write value should be 0. | R/W |
| b2 | LVD1RF | Voltage Monitoring 1 Reset Detect Flag | 0: Voltage monitoring 1 reset not detected.<br>1: Voltage monitoring 1 reset detected. | R/(W) *1 |
| b3 | LVD2RF | Voltage Monitoring 2 Reset Detect Flag | 0: Voltage monitoring 2 reset not detected.<br>1: Voltage monitoring 2 reset detected. | R/(W) *1 |
| b7 to b4 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |

Note 1. Only 0 can be written to clear the flag.

#### PORF Flag (Power-On Reset Detect Flag)

The PORF flag indicates that a power-on reset has occurred.

[Setting condition]

- When a power-on reset occurs.

[Clearing conditions]

- When a reset shown in Table 6.2 occurs.
- When PORF is read as 1 and then 0 is written to PORF.

#### LVD1RF Flag (Voltage Monitoring 1 Reset Detect Flag)

The LVD1RF flag indicates that VCC voltage has fallen below Vdet1.

[Setting condition]

- When Vdet1-level VCC voltage is detected.

[Clearing conditions]

- When a reset listed in Table 6.2 occurs.
- When LVD1RF is read as 1 and then 0 is written to LVD1RF.

#### LVD2RF Flag (Voltage Monitoring 2 Reset Detect Flag)

The LVD2RF flag indicates that VCC voltage has fallen below Vdet2.

[Setting condition]

- When Vdet2-level VCC voltage is detected.

[Clearing conditions]

- When a reset listed in Table 6.2 occurs.
- When LVD2RF is read as 1 and then 0 is written to LVD2RF.

## 6.2.2 Reset Status Register 1 (RSTSR1)

Address(es): 0008 C291h

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | CWSF |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0/1*1 |

Note 1. The value after reset depends on the reset source.

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b0 | CWSF | Cold/Warm Start Determination Flag | 0: Cold start<br>1: Warm start | R/(W) *1 |
| b7 to b1 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |

Note 1. Only 1 can be written to set the flag.

RSTSR1 determines whether a power-on reset has caused the reset processing (cold start) or a reset signal input during operation has caused the reset processing (warm start).

**CWSF Flag (Cold/Warm Start Determination Flag)**

The CWSF flag indicates the type of reset processing: cold start or warm start.
The CWSF flag is initialized at a power-on.
[Setting condition]

- When 1 is written through programming; it is not set to 0 even when 0 is written.

[Clearing condition]

- When a reset listed in Table 6.2 occurs.

## 6.2.3 Reset Status Register 2 (RSTSR2)

Address(es): 0008 00C0h

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | SWRF | — | IWDTRF |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0*1 | 0 | 0*1 |

Note 1. The value after reset depends on the reset source.

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b0 | IWDTRF | Independent Watchdog Timer Reset Detect Flag | 0: Independent watchdog timer reset not detected.<br>1: Independent watchdog timer reset detected. | R/(W) *1 |
| b1 | — | Reserved | This bit is read as 0. The write value should be 0. | R/W |
| b2 | SWRF | Software Reset Detect Flag | 0: Software reset not detected.<br>1: Software reset detected. | R/(W) *1 |
| b7 to b3 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |

Note 1. Only 0 can be written to clear the flag.

### IWDTRF Flag (Independent Watchdog Timer Reset Detect Flag)

The IWDTRF flag indicates that an independent watchdog timer reset has occurred.

[Setting condition]

- When an independent watchdog timer reset occurs.

[Clearing conditions]

- When a reset listed in Table 6.2 occurs.
- When IWDTRF is read as 1 and then 0 is written to IWDTRF.

### SWRF Flag (Software Reset Detect Flag)

The SWRF flag indicates that a software reset has occurred.

[Setting condition]

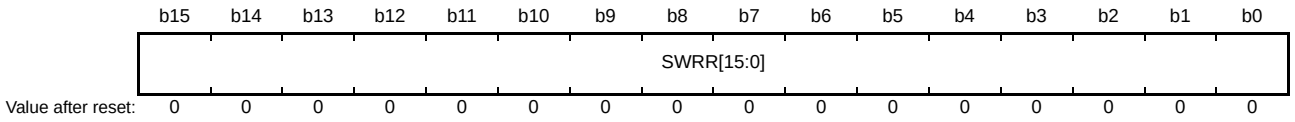
- When a software reset occurs.

[Clearing conditions]

- When a reset listed in Table 6.2 occurs.
- When SWRF is read as 1 and then 0 is written to SWRF.

## 6.2.4 Software Reset Register (SWRR)

Address(es): 0008 00C2h




| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b15 to b0 | SWRR[15:0] | Software Reset | Writing A501h resets the MCU. These bits are read as 0000h. | R/W |

Note: •Set the PRCR.PRC1 bit to 1 (write enabled) before rewriting this register.

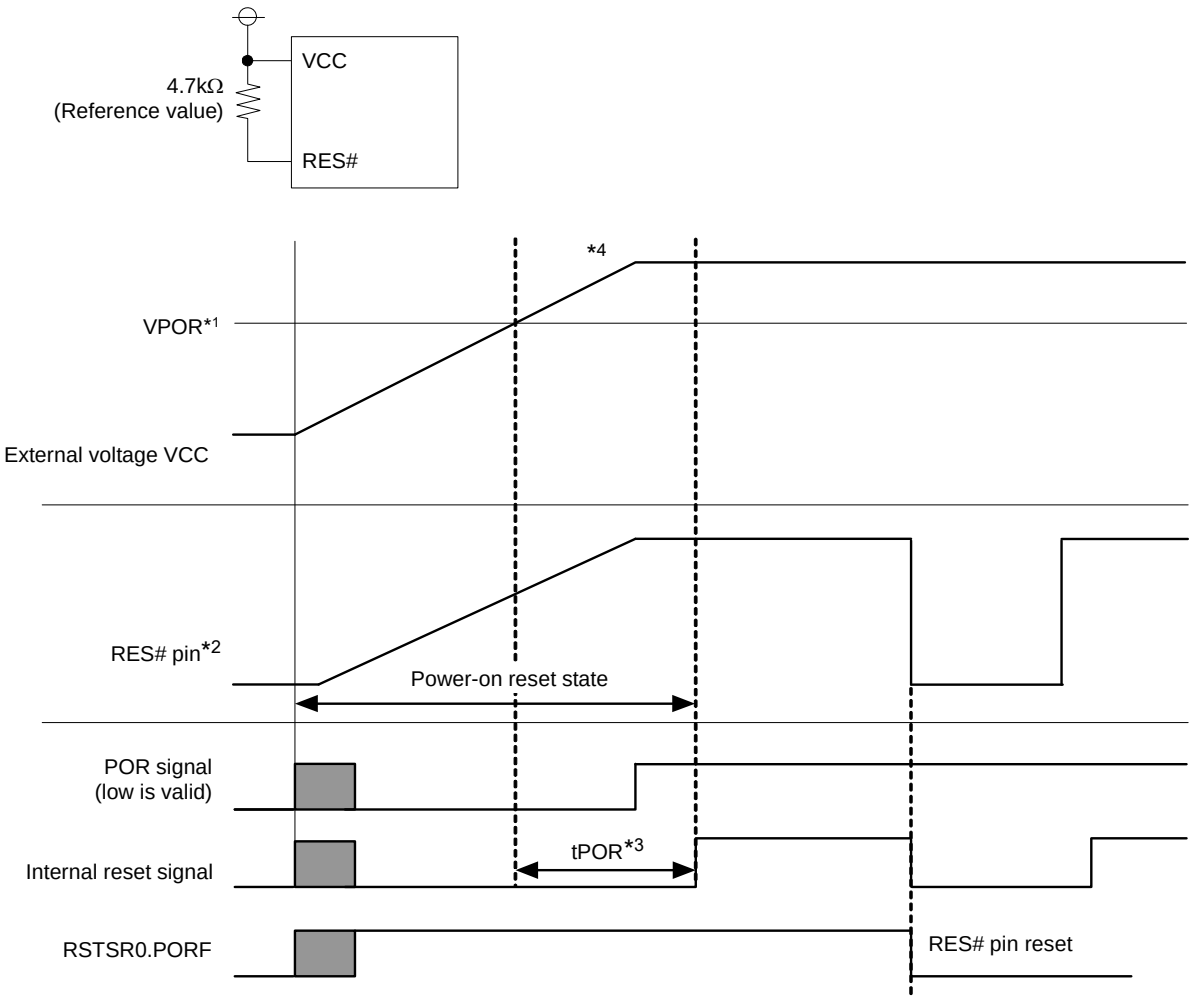
## 6.3 Operation

### 6.3.1 RES# Pin Reset

This is a reset generated by the RES# pin.
When the RES# pin is driven low, all the processing in progress is aborted and the MCU enters a reset state.
To ensure the MCU is reset, the RES# pin should be held low for the specified power supply stabilization time at a power-on.
When the RES# pin is driven high from low, the internal reset is canceled after the RES# post-cancelation wait time (tRESWT) has elapsed, and then the CPU starts the reset exception handling.
For details, refer to section 36, Electrical Characteristics.

### 6.3.2 Power-On Reset

The power-on reset is an internal reset generated by the power-on reset circuit.
A power-on reset is generated when power is supplied to the RES# pin while it is connected to VCC via a resistor. When connecting a capacitor to the RES# pin, also ensure that the voltage on the RES# pin is always at least VIH. For details on VIH, refer to section 36, Electrical Characteristics. After VCC has exceeded VPOR and the specified period (power-on reset time) has elapsed, the internal reset is canceled and the CPU starts the reset exception handling. The power-on reset time is a stabilization period for the external power supply and the MCU circuit. After a power-on reset has been generated, the RSTSR0.PORF flag is set to 1. The PORF flag is initialized by a RES# pin reset.
Figure 6.1 shows an example of the power-on reset circuit and its operation.




Note: • For details on the electrical characteristics, refer to section 32, Electrical Characteristics.
Note 1. VPOR shows the power-on reset detection level.
Note 2. Ensure that the voltage on the RES# pin is always at least VIH.
Note 3. tPOR shows the period for power-on reset.
Note 4. At the power-on, VCC must rise to at least the minimum guaranteed voltage before the POR reset is released.

**Figure 6.1 Operation Examples during Power-On Reset**

### 6.3.3 Voltage Monitoring 1 Reset and Voltage Monitoring 2 Reset

The voltage monitoring 1 reset and voltage monitoring 2 reset are internal resets generated by the voltage monitoring circuit.

When the voltage monitoring 1 interrupt/reset enable bit (LVD1RIE) is set to 1 (enabled) and the voltage monitoring 1 circuit mode select bit (LVD1RI) is set to 1 (voltage monitoring 1 reset enabled when the voltage falls to and below Vdet1) in voltage monitoring 1 circuit control register 0 (LVD1CR0), the RSTSR0.LVD1RF flag is set to 1 and the voltage-detection circuit generates a voltage monitoring 1 reset if VCC falls to or below Vdet1.

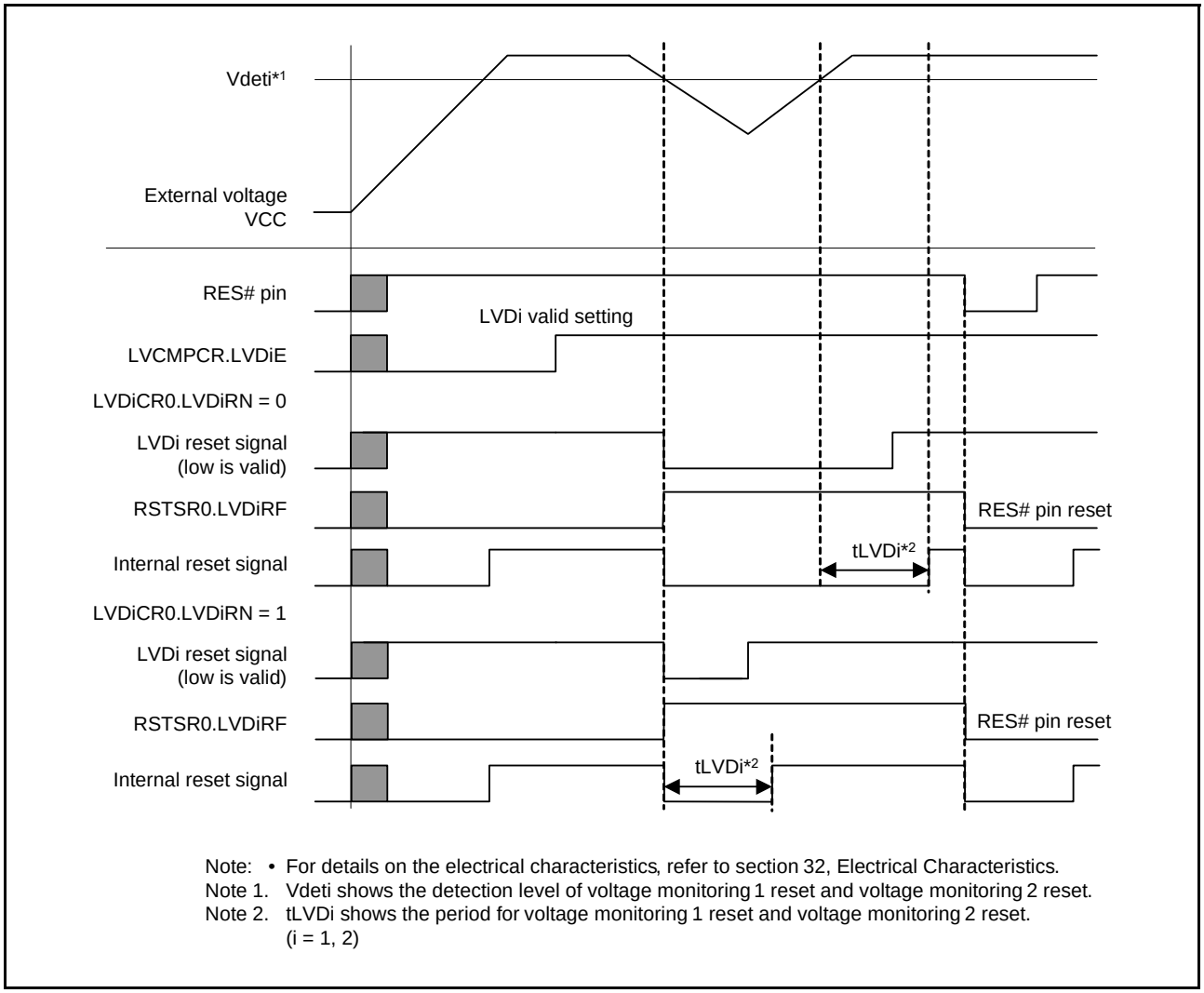
Likewise, when the voltage monitoring 2 interrupt/reset enable bit (LVD2RIE) is set to 1 (enabled) and the voltage monitoring 2 circuit mode select bit (LVD2RI) is set to 1 (voltage monitoring 2 reset enabled when the voltage falls to and below Vdet2) in voltage monitoring 2 circuit control register 0 (LVD2CR0), the RSTSR0.LVD2RF flag is set to 1 and the voltage detection circuit generates a voltage monitoring 2 reset if VCC falls to or below Vdet2.

Timing for release from the voltage monitoring 1 reset state is selectable with the voltage monitoring 1 reset negate select bit (LVD1RN) in the LVD1CR0 register. When the LVD1CR0.LVD1RN bit is 0 and VCC has fallen to or below Vdet1, the CPU is released from the internal reset state and starts reset exception handling once the LVD1 reset time (tLVD1) has elapsed after VCC has risen above Vdet1. When the LVD1CR0.LVD1RN bit is 1 and VCC has fallen to or below Vdet1, the CPU is released from the internal reset state and starts reset exception handling once the LVD1 reset time (tLVD1) has elapsed.

Likewise, timing for release from the voltage monitoring 2 reset state is selectable by setting the voltage monitoring 2 reset negate select bit (LVD2RN) in the LVD2CR0 register. Detection levels Vdet1 and Vdet2 can be changed by settings in the voltage detection level select register (LVDLVLR).

Figure 6.2 shows examples of operations during voltage monitoring 1 and 2 resets.

For details on the voltage monitoring 1 reset and voltage monitoring 2 reset, refer to section 8, Voltage Detection Circuit (LVDAa).




Note: • For details on the electrical characteristics, refer to section 32, Electrical Characteristics.
Note 1. Vdeti shows the detection level of voltage monitoring 1 reset and voltage monitoring 2 reset.
Note 2. tLVDi shows the period for voltage monitoring 1 reset and voltage monitoring 2 reset.
(i = 1, 2)

**Figure 6.2 Operation Examples during Voltage Monitoring 1 Reset and Voltage Monitoring 2 Reset**

### 6.3.4 Independent Watchdog Timer Reset

The independent watchdog timer reset is an internal reset generated by the independent watchdog timer.
Enabling and disabling output of the independent watchdog timer reset from the independent watchdog timer can be selected by setting the IWDT reset control register (IWDTRCR) and option function select register 0 (OFS0).
When output of the independent watchdog timer reset is enabled, an independent watchdog timer reset is generated if the independent watchdog timer underflows, or if data is written outside the refresh-permitted period. After the independent watchdog timer reset has been generated and the internal reset time (tRESW2) has elapsed, the internal reset is canceled and the CPU starts the reset exception handling.
For details on the independent watchdog timer reset, refer to section 24, Independent Watchdog Timer (IWDTa).

### 6.3.5 Software Reset

The software reset is an internal reset generated by the software reset circuit.
A software reset is generated when A501h is written to the SWRR register. After the software reset has been generated and tRESW2 has elapsed, the internal reset is canceled and the CPU starts the reset exception handling.

### 6.3.6 Determination of Cold/Warm Start

By reading the RSTSR1.CWSF flag, the type of reset processing caused can be identified; that is, whether a power-on reset has caused the reset processing (cold start) or a reset signal input during operation has caused the reset processing (warm start).
The RSTSR1.CWSF flag is set to 0 when a power-on reset occurs (cold start); otherwise the flag is not set to 0. The flag is set to 1 when 1 is written to it through programming; it is not set to 0 even when 0 is written.
Figure 6.3 shows an example of cold/warm start determination operation.

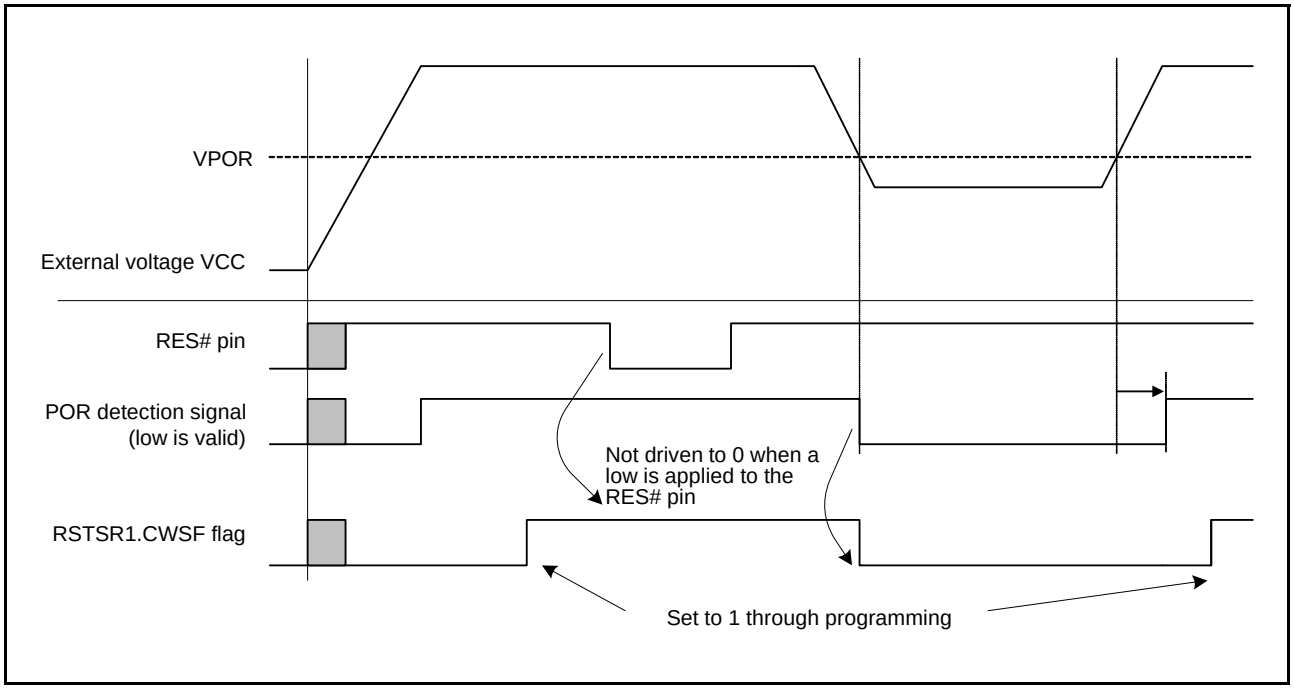



**Figure 6.3 Example of Cold/Warm Start Determination Operation**

### 6.3.7 Determination of Reset Generation Source

Reading the RSTSR0 and RSTSR2 registers determines which reset was used to execute the reset exception handling. Figure 6.4 shows an example of the flow to identify a reset generation source.

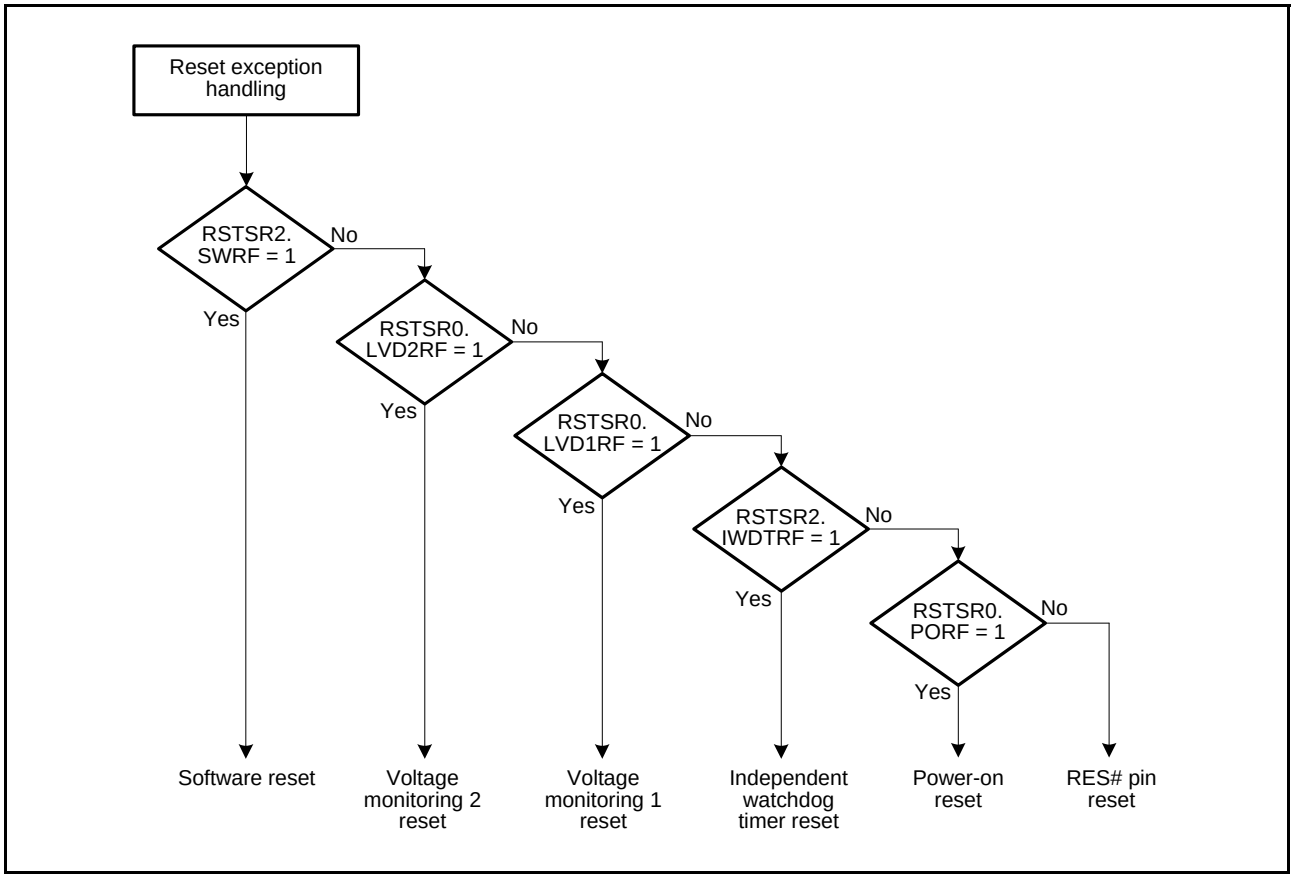



**Figure 6.4 Example of Reset Generation Source Determination Flow**

# 7. Option-Setting Memory

## 7.1 Overview

Option-setting memory refers to a set of registers that are provided for selecting the state of the microcontroller after a reset. The option-setting memory is allocated in the ROM.
Figure 7.1 shows the option-setting memory area.

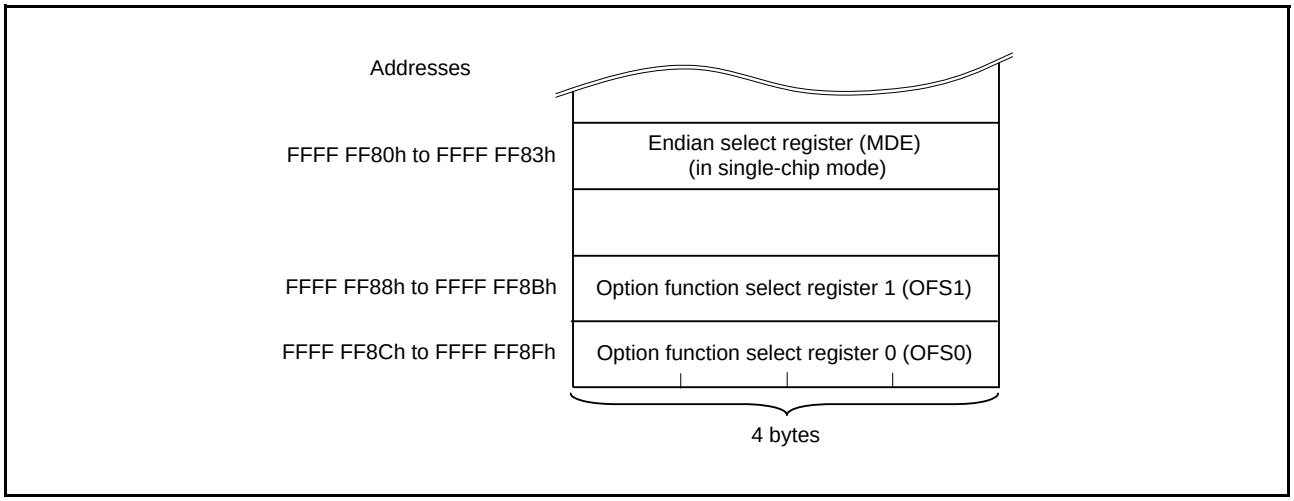



**Figure 7.1 Option-Setting Memory Area**

## 7.2 Register Descriptions

### 7.2.1 Option Function Select Register 0 (OFS0)

Address(es): FFFF FF8Ch

| | b31 | b30 | b29 | b28 | b27 | b26 | b25 | b24 | b23 | b22 | b21 | b20 | b19 | b18 | b17 | b16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset: | The value set by the user*1 | | | | | | | | | | | | | | | |

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | IWDTSLCSTP | — | IWDTRSTIRQS | IWDTRPSS[1:0] | | IWDTRPES[1:0] | | IWDTCKS[3:0] | | | | IWDTTOPS[1:0] | | IWDTSTRT | — |
| Value after reset: | The value set by the user*1 | | | | | | | | | | | | | | | |

Note 1. The value of the blank product is FFFF FFFFh. It is set to the written value after written by the user.

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b0 | — | Reserved | When reading, this bit returns the value written by the user. The write value should be 1. | R |
| b1 | IWDTSTRT | IWDT Start Mode Select | 0: IWDT is automatically activated in auto-start mode after a reset<br>1: IWDT is halted after a reset | R |
| b3, b2 | IWDTTOPS[1:0] | IWDT Timeout Period Select | b3 b2<br>0 0: 128 cycles (03FFh)<br>0 1: 512 cycles (0FFFh)<br>1 0: 1024 cycles (1FFFh)<br>1 1: 2048 cycles (3FFFh) | R |
| b7 to b4 | IWDTCKS[3:0] | IWDT Clock Frequency Division Ratio Select | b7 b4<br>0 0 0 0: ×1 (Cycle period: 136 ms)<br>0 0 1 0: ×1/16 (Cycle period: 2.18 s)<br>0 0 1 1: ×1/32 (Cycle period: 4.36 s)<br>0 1 0 0: ×1/64 (Cycle period: 8.73 s)<br>1 1 1 1: ×1/128 (Cycle period: 17.5 s)<br>0 1 0 1: ×1/256 (Cycle period: 34.9 s)<br>Settings other than above are prohibited. | R |
| b9, b8 | IWDTRPES[1:0] | IWDT Window End Position Select | b9 b8<br>0 0: 75%<br>0 1: 50%<br>1 0: 25%<br>1 1: 0% (No window end position setting) | R |
| b11, b10 | IWDTRPSS[1:0] | IWDT Window Start Position Select | b11 b10<br>0 0: 25%<br>0 1: 50%<br>1 0: 75%<br>1 1: 100% (No window start position setting) | R |
| b12 | IWDTRSTIRQS | IWDT Reset Interrupt Request Select | 0: Non-maskable interrupt request is enabled<br>1: Reset is enabled | R |
| b13 | — | Reserved | When reading, this bit returns the value written by the user. The write value should be 1. | R |
| b14 | IWDTSLCSTP | IWDT Sleep Mode Count Stop Control | 0: Counting stop is disabled<br>1: Counting stop is enabled when entering sleep, software standby mode, and deep sleep mode | R |
| b31 to b15 | — | Reserved | When reading, these bits return the value written by the user. The write value should be 1. | R |

The OFS0 register selects the operations of the independent watchdog timer (IWDT) after a reset.
The OFS0 register is allocated in the ROM. Set this register at the same time as writing the program. After writing to the OFS0 register once, do not write to it again.
When erasing the block including the OFS0 register, the OFS0 register value becomes FFFF FFFFh.

The setting in the OFS0 register is ineffective in boot mode.

**IWDTSTRT Bit (IWDT Start Mode Select)**

This bit selects the mode in which the IWDT is activated after a reset (stopped state or activated in auto-start mode). When activated in auto-start mode, the OFS0 register setting for the IWDT is effective.

**IWDTTOPS[1:0] Bits (IWDT Timeout Period Select)**

These bits select the timeout period, i.e. the time it takes for the down-counter to underflow, as 128, 512, 1024, or 2048 cycles of the frequency-divided clock set by the IWDTCKS[3:0] bits. The time (number of clock cycles for the IWDT) it takes to underflow after a refresh operation is determined by the combination of the IWDTCKS[3:0] bits and IWDTTOPS[1:0] bits.

For details, refer to section 24, Independent Watchdog Timer (IWDTa).

**IWDTCKS[3:0] Bits (IWDT Clock Frequency Division Ratio Select)**

These bits select, from 1/1, 1/16, 1/32, 1/64, 1/128, and 1/256, the division ratio of the prescaler to divide the frequency of the clock for the IWDT. Using the setting of these bits together with the IWDTTOPS[1:0] bit setting, the IWDT counting period can be set from 128 to 524288 clock cycles for the IWDT.

For details, refer to section 24, Independent Watchdog Timer (IWDTa).

**IWDTRPES[1:0] Bits (IWDT Window End Position Select)**

These bits select the position of the end of the window for the down-counter as 0%, 25%, 50%, or 75% of the value being counted by the counter. The value of the window end position must be smaller than the value of the window start position (window start position > window end position). If the value for the window end position is greater than the value for the window start position, only the value for the window start position is effective.

The counter values corresponding to the settings for the start and end positions of the window in the IWDTRPSS[1:0] and IWDTRPES[1:0] bits vary with the setting of the IWDTTOPS[1:0] bits.

For details, refer to section 24, Independent Watchdog Timer (IWDTa).

**IWDTRPSS[1:0] Bits (IWDT Window Start Position Select)**

These bits select the position where the window for the down-counter starts as 25%, 50%, 75%, or 100% of the value being counted (the point at which counting starts is 100% and the point at which an underflow occurs is 0%). The interval between the positions where the window starts and ends becomes the period in which refreshing is possible, and refreshing is not possible outside this period.

For details, refer to section 24, Independent Watchdog Timer (IWDTa).

**IWDTRSTIRQS Bit (IWDT Reset Interrupt Request Select)**

The setting of this bit selects the operation on an underflow of the down-counter or generation of a refresh error. Either an independent watchdog timer reset or a non-maskable interrupt request is selectable.

For details, refer to section 24, Independent Watchdog Timer (IWDTa).

**IWDTSLCSTP Bit (IWDT Sleep Mode Count Stop Control)**

This bit selects to stop counting when entering sleep, software standby mode, and deep sleep mode.

For details, refer to section 24, Independent Watchdog Timer (IWDTa).

## 7.2.2 Option Function Select Register 1 (OFS1)

Address(es): FFFF FF88h

| | b31 | b30 | b29 | b28 | b27 | b26 | b25 | b24 | b23 | b22 | b21 | b20 | b19 | b18 | b17 | b16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Value after reset: | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| | The value set by the user*1 | | | | | | | | | | | | | | | |

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Value after reset: | — | — | — | — | — | — | — | HOCOEN | STUPLVD1LVL[3:0] | | | | — | — | STUPLVD1REN | FASTSTUP |
| | The value set by the user*1 | | | | | | | | | | | | | | | |

Note 1. The value of the blank product is FFFF FFFFh. It is set to the written value after written by the user.

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b0 | FASTSTUP | Power-On Fast Startup Time | 0: Fast startup time at power on<br>1: Normal startup | R |
| b1 | STUPLVD1REN | Startup Voltage Monitoring 1 Reset Enable | 0: Voltage monitoring 1 reset is enabled at startup<br>1: Voltage monitoring 1 reset is disabled at startup | R |
| b3, b2 | — | Reserved | When reading, these bits return the value written by the user. The write value should be 1. | R |
| b7 to b4 | STUPLVD1LVL[3:0] | Startup Voltage Monitoring 1 Reset Detection Level Select | b7 b4<br>0 1 0 0: 3.10 V<br>0 1 0 1: 3.00 V<br>0 1 1 0: 2.90 V<br>0 1 1 1: 2.79 V<br>1 0 0 0: 2.68 V<br>1 0 0 1: 2.58 V<br>1 0 1 0: 2.48 V<br>1 0 1 1: 2.06 V<br>1 1 0 0: 1.96 V<br>1 1 0 1: 1.86 V<br>Settings other than above are prohibited when the STUPLVD1REN bit is 0. | R |
| b8 | HOCOEN | HOCO Oscillation Enable | 0: HOCO oscillation is enabled after a reset<br>1: HOCO oscillation is disabled after a reset | R |
| b31 to b9 | — | Reserved | When reading, these bits return the value written by the user. The write value should be 1. | R |

The OFS1 register is allocated in the ROM. Set this register at the same time as writing the program. After writing, do not write additions to this register.
When erasing the block including the OFS1 register, the setting in the OFS1 register is ineffective, and the OFS1 register value becomes FFFF FFFFh.
The setting in the OFS1 register is ineffective in boot mode.

### FASTSTUP Bit (Power-On Fast Startup Time)

The startup time can be reduced by setting this bit to 0 (fast startup time at power on) when it is possible to meet the power-on VCC rising gradient (during fast startup time) shown in Electrical Characteristics.
Do not set this bit to 0 when it is not possible to meet the power-on VCC rising gradient (during fast startup time).

### STUPLVD1REN Bit (Startup Voltage Monitoring 1 Reset Enable)

This bit selects whether the voltage monitoring 1 reset is enabled or disabled after a reset.
The Vdet1 voltage to be monitored by the voltage detection 1 circuit is selected by the STUPLVD1LVL[3:0] bits.
When this bit is set to 0 (voltage monitoring 1 reset is enabled at startup) and the power is turned on, the power-on VCC rising gradient specification is the power-on VCC rising gradient (when voltage monitoring 1 reset is enabled at startup) shown in section 36, Electrical Characteristics, and there is no specified maximum value.
When the STUPLVD1REN bit is set to 0, the MCU starts up with voltage monitoring 1 reset enabled, regardless of the FASTSTUP bit setting.

### STUPLVD1LVL[3:0] Bits (Startup Voltage Monitoring 1 Reset Detection Level Select)

These bits select the voltage detection level to be monitored by the voltage detection 1 circuit when the STUPLVD1REN bit is set to 0.

### HOCOEN Bit (HOCO Oscillation Enable)

This bit selects whether the HOCO oscillation enable bit is enabled or disabled after a reset.
Setting the HOCOEN bit to 0 allows the HOCO oscillation to be started before the CPU starts operation, and therefore reduces the waiting time for oscillation stabilization.
Note that even if the HOCOEN bit is set to 0, the system clock source is not switched to HOCO. The system clock source is switched to HOCO only by modifying the clock source select bits (SCKCR3.CKSEL[2:0]) from the CPU.
Also, when the OFS1.HOCOEN bit is set to 0, the HOCO oscillation stabilization time (tHOCO) is secured by hardware, so the clock with the accuracy of the HOCO oscillation frequency (fHOCO) shown in Electrical Characteristics is supplied after release from the CPU reset state.

## 7.2.3 Endian Select Register (MDE)

Address(es): FFFF FF80h: MDE (in single-chip mode)

| | b31 | b30 | b29 | b28 | b27 | b26 | b25 | b24 | b23 | b22 | b21 | b20 | b19 | b18 | b17 | b16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset: | The value set by the user*1 | | | | | | | | | | | | | | | |

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | MDE[2:0] | | |
| Value after reset: | The value set by the user*1 | | | | | | | | | | | | | | | |

Note 1. The value of the blank product is FFFF FFFFh. It is set to the written value after written by the user.

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b2 to b0 | MDE[2:0] | Endian Select | b2 b0<br>0 0 0: Big endian<br>1 1 1: Little endian<br>Settings other than above are prohibited. | R |
| b31 to b3 | — | Reserved | When reading, these bits return the value written by the user.<br>The write value should be 1. | R |

The MDE register selects the endian for the CPU. The endian select register (MDE) at address FFFF FF80h is used to select the endian.
MDE is allocated in the ROM. Set the register at the same time as writing the program. After writing to the register once, do not write to it again.
When erasing the block including the MDE register, the MDE register value becomes FFFF FFFFh.

**MDE[2:0] Bits (Endian Select)**

These bits select little endian or big endian for the CPU.

## 7.3 Usage Note

### 7.3.1 Setting Example of Option-Setting Memory

Since the option-setting memory is allocated in the ROM, values cannot be written by executing instructions. Write appropriate values when writing the program. An example of the settings is shown below.

- To set ffff fff8h in the OFS0 register
  .org 0ffff ff8ch
  .lword 0ffffff8h

Note: • Programming formats vary depending on the compiler. Refer to the compiler manual for details.

# 8. Voltage Detection Circuit (LVDAa)

The voltage detection circuit (LVD) monitors the voltage level input to the VCC pin using a program.

## 8.1 Overview

In voltage detection 1, the detection voltage can be selected from 10 levels using the voltage detection level select register (LVDLVLR).
In voltage detection 2, the detection voltage can be selected from four levels by switching between input voltages to VCC and the CMPA2 pin.
Reset/interrupt of voltage monitoring 1, and reset/interrupt of voltage monitoring 2 can be used.

Table 8.1 lists the specifications of the voltage detection circuit. Figure 8.1 is a block diagram of the voltage detection circuit. Figure 8.2 is a block diagram of the voltage monitoring 1 interrupt/reset circuit. Figure 8.3 is a block diagram of the voltage monitoring 2 interrupt/reset circuit.

**Table 8.1 LVD Specifications**

| Item | | Voltage Monitoring 1 | Voltage Monitoring 2 |
|---|---|---|---|
| VCC monitoring | Monitored voltage | Vdet1 | Vdet2 |
| | Detection target | When voltage rises above or drops below Vdet1 | When voltage rises above or drops below Vdet2 |
| | | | Input voltages to VCC and the CMPA2 pin can be switched using the LVCMPCR.EXVCCINP2 bit |
| | Detection voltage | Voltage selectable from 10 levels using the LVDLVLR.LVD1LVL[3:0] bits | Voltage selectable from four levels using the LVDLVLR.LVD2LVL[1:0] bits |
| | Monitoring flag | LVD1SR.LVD1MON flag: Monitors whether voltage is higher or lower than Vdet1 | LVD2SR.LVD2MON flag: Monitors whether voltage is higher or lower than Vdet2 |
| | | LVD1SR.LVD1DET flag: Vdet1 passage detection | LVD2SR.LVD2DET flag: Vdet2 passage detection |
| Process upon voltage detection | Reset | Voltage monitoring 1 reset | Voltage monitoring 2 reset |
| | | Reset when Vdet1 > VCC<br>CPU restart timing selectable: after specified time with VCC > Vdet1 or Vdet1 > VCC | Reset when Vdet2 > VCC or the CMPA2 pin<br>CPU restart timing selectable: after specified time with VCC or the CMPA2 pin > Vdet2 or after specified time with Vdet2 > VCC or the CMPA2 pin |
| | Interrupt | Voltage monitoring 1 interrupt | Voltage monitoring 2 interrupt |
| | | Non-maskable or maskable interrupt is selectable | Non-maskable or maskable interrupt is selectable |
| | | Interrupt request issued when Vdet1 > VCC and VCC > Vdet1 or either | Interrupt request issued when Vdet2 > VCC or the CMPA2 pin and VCC or the CMPA2 pin > Vdet2 or either |
| Event link function | | Available<br>Vdet1 passage detection event output | Not available |

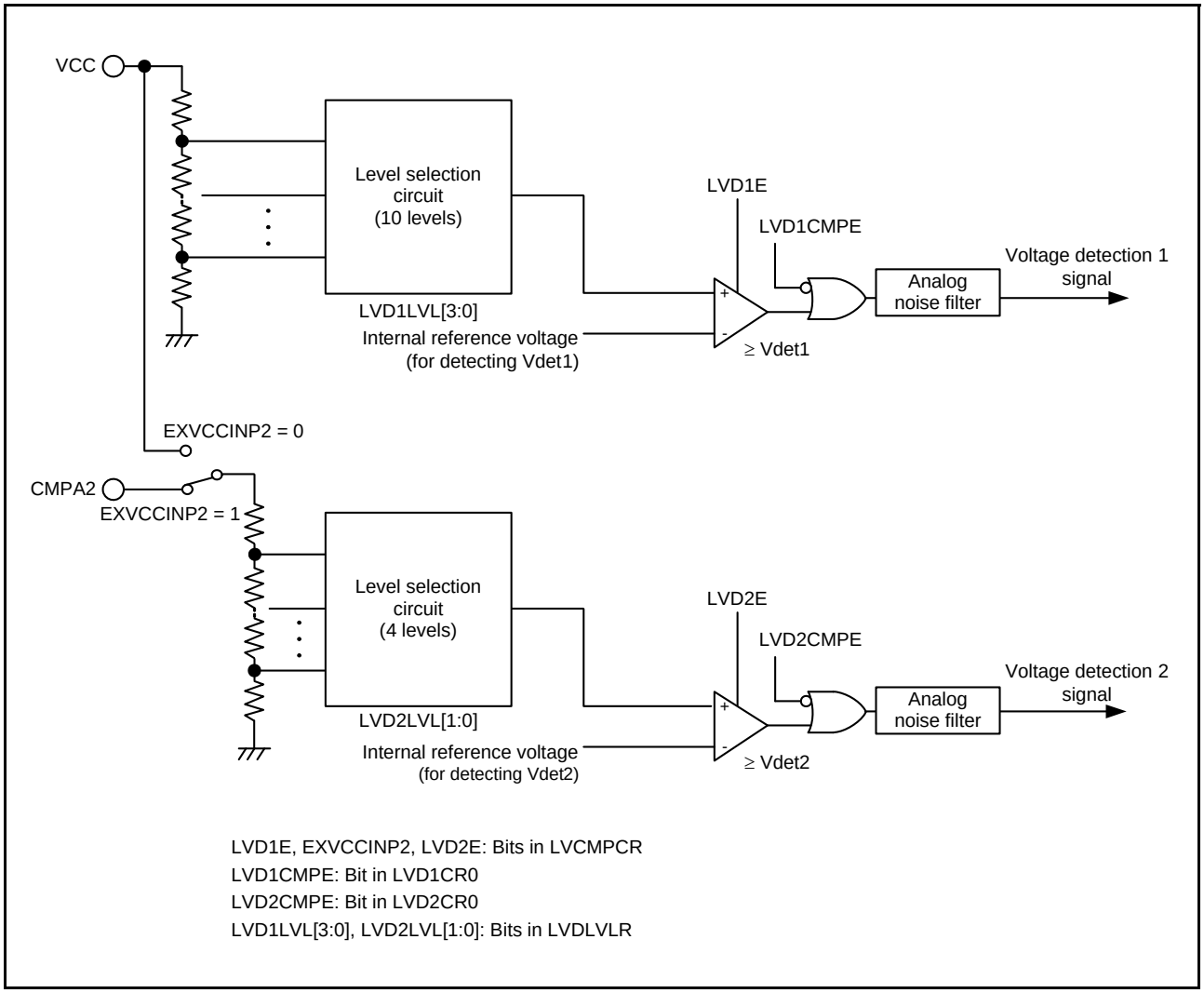



**Figure 8.1 Block Diagram of the LVD**

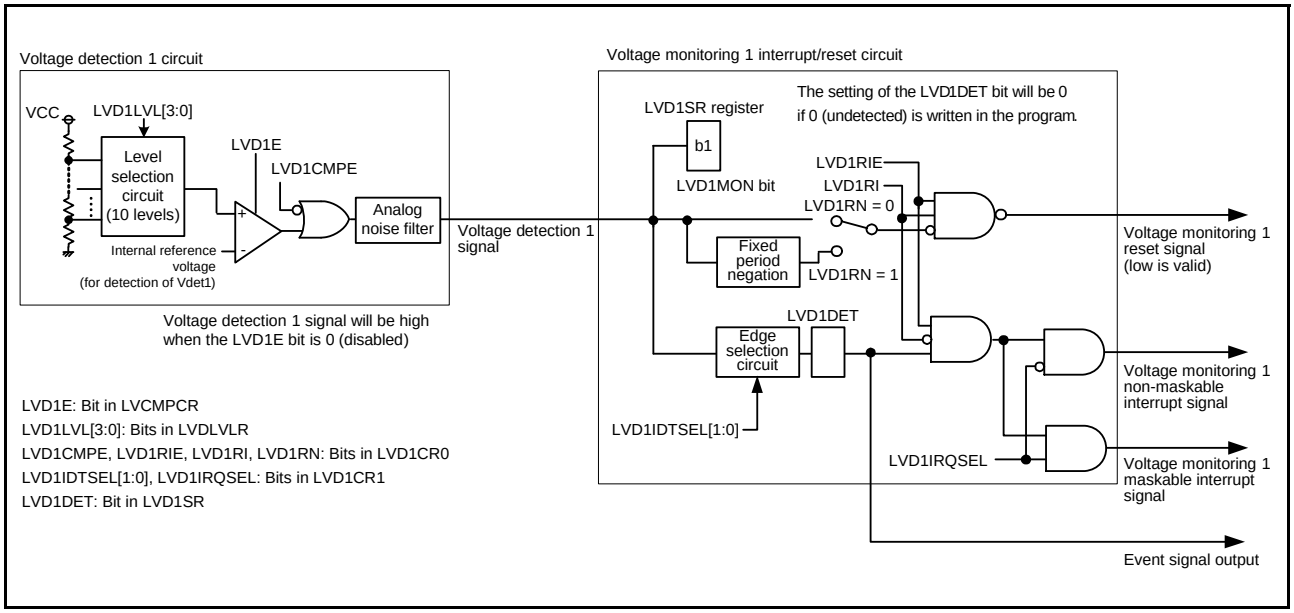



**Figure 8.2 Block Diagram of Voltage Monitoring 1 Interrupt/Reset Circuit**

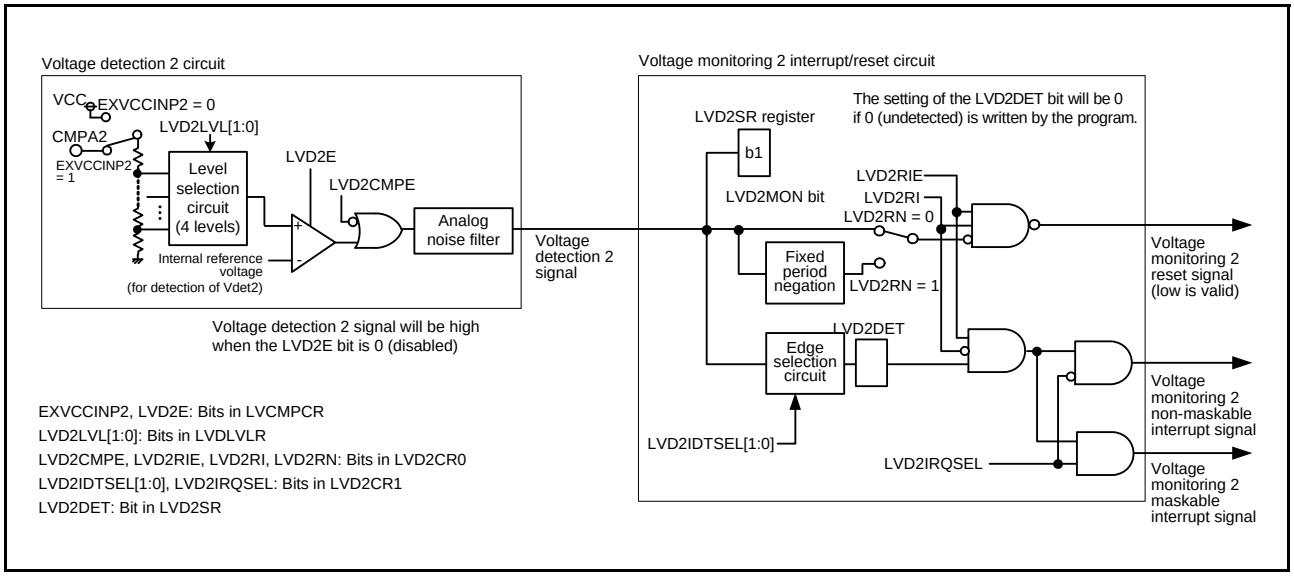



**Figure 8.3 Block Diagram of Voltage Monitoring 2 Interrupt/Reset Circuit**

Table 8.2 lists the I/O pins relevant to the voltage detection circuit.

**Table 8.2 I/O Pins of the Voltage Detection Circuit**

| Pin Name | I/O | Function |
|---|---|---|
| CMPA2 | Input | Detection target voltage pin for voltage detection 2 |

## 8.2 Register Descriptions

### 8.2.1 Voltage Monitoring 1 Circuit Control Register 1 (LVD1CR1)

Address(es): 0008 00E0h

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | LVD1IRQSEL | LVD1IDTSEL[1:0] | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b1, b0 | LVD1IDTSEL[1:0] | Voltage Monitoring 1 Interrupt ELC Event Generation Condition Select | b1 b0<br>0 0: When VCC ≥ Vdet1 (rise) is detected<br>0 1: When VCC < Vdet1 (drop) is detected<br>1 0: When drop and rise are detected<br>1 1: Setting prohibited | R/W |
| b2 | LVD1IRQSEL | Voltage Monitoring 1 Interrupt Type Select | 0: Non-maskable interrupt<br>1: Maskable interrupt | R/W |
| b7 to b3 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |

Note: •Set the PRCR.PRC3 bit to 1 (write enabled) before rewriting this register.

## 8.2.2 Voltage Monitoring 1 Circuit Status Register (LVD1SR)

Address(es): 0008 00E1h

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | LVD1MON | LVD1DET |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b0 | LVD1DET | Voltage Monitoring 1 Voltage Change Detection Flag | 0: Not detected<br>1: Vdet1 passage detection | R/(W) *1 |
| b1 | LVD1MON | Voltage Monitoring 1 Signal Monitor Flag | 0: VCC < Vdet1<br>1: VCC ≥ Vdet1 or LVD1MON circuit is disabled | R |
| b7 to b2 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |

Note: •Set the PRCR.PRC3 bit to 1 (write enabled) before rewriting this register.
Note 1. Only 0 can be written to this bit. After writing 0 to this bit, it takes two system clock cycles for the bit to be read as 0.

### LVD1DET Flag (Voltage Monitoring 1 Voltage Change Detection Flag)

The LVD1DET flag is enabled when the LVCMPCR.LVD1E bit is 1 (voltage detection 1 circuit enabled) and the LVD1CR0.LVD1CMPE bit is 1 (voltage monitoring 1 circuit comparison result output enabled).
The LVD1DET flag should be set to 0 after LVD1CR0.LVD1RIE is set to 0 (disabled). LVD1CR0.LVD1RIE can be set to 1 (enabled) again after a period of two or more cycles of PCLKB has elapsed.
Depending on the number of cycles of PCLKB defined for access to read an I/O register, two or more cycles of PCLKB may have to be secured as waiting time.

### LVD1MON Flag (Voltage Monitoring 1 Signal Monitor Flag)

The LVD1MON flag is enabled when the LVCMPCR.LVD1E bit is 1 (voltage detection 1 circuit enabled) and the LVD1CR0.LVD1CMPE bit is 1 (voltage monitoring 1 circuit comparison result output enabled).

## 8.2.3 Voltage Monitoring 2 Circuit Control Register 1 (LVD2CR1)

Address(es): 0008 00E2h

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | LVD2IRQSEL | LVD2IDTSEL [1:0] | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b1, b0 | LVD2IDTSEL [1:0] | Voltage Monitoring 2 Interrupt Generation Condition Select | b1 b0<br>0 0: When VCC or the CMPA2 pin ≥ Vdet2 (rise) is detected<br>0 1: When VCC or the CMPA2 pin < Vdet2 (drop) is detected<br>1 0: When drop and rise are detected<br>1 1: Setting prohibited | R/W |
| b2 | LVD2IRQSEL | Voltage Monitoring 2 Interrupt Type Select | 0: Non-maskable interrupt<br>1: Maskable interrupt | R/W |
| b7 to b3 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |

Note: •Set the PRCR.PRC3 bit to 1 (write enabled) before rewriting this register.

## 8.2.4 Voltage Monitoring 2 Circuit Status Register (LVD2SR)

Address(es): 0008 00E3h

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | LVD2MON | LVD2DET |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b0 | LVD2DET | Voltage Monitoring 2 Voltage Change Detection Flag | 0: Not detected<br>1: Vdet2 passage detection | R/(W) *1 |
| b1 | LVD2MON | Voltage Monitoring 2 Signal Monitor Flag | 0: VCC or the CMPA2 pin < Vdet2<br>1: VCC or the CMPA2 pin ≥ Vdet2 or LVD2MON is disabled | R |
| b7 to b2 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |

Note: •Set the PRCR.PRC3 bit to 1 (write enabled) before rewriting this register.
Note 1. Only 0 can be written to this bit. After writing 0 to this bit, it takes two system clock cycles for the bit to be read as 0.

### LVD2DET Flag (Voltage Monitoring 2 Voltage Change Detection Flag)

The LVD2DET flag is enabled when the LVCMPCR.LVD2E bit is 1 (voltage detection 2 circuit enabled) and the LVD2CR0.LVD2CMPE bit is 1 (voltage monitoring 2 circuit comparison result output enabled).
The LVD2DET flag should be set to 0 after LVD2CR0.LVD2RIE is set to 0 (disabled). LVD2CR0.LVD2RIE can be set to 1 (enabled) again after a period of two or more cycles of PCLKB has elapsed.
Depending on the number of cycles of PCLKB defined for access to read an I/O register, two or more cycles of PCLKB may have to be secured as waiting time.

### LVD2MON Flag (Voltage Monitoring 2 Signal Monitor Flag)

The LVD2MON flag is enabled when the LVCMPCR.LVD2E bit is 1 (voltage detection 2 circuit enabled) and the LVD2CR0.LVD2CMPE bit is 1 (voltage monitoring 2 circuit comparison result output enabled).

## 8.2.5 Voltage Monitoring Circuit Control Register (LVCMPCR)

Address(es): 0008 C297h

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | LVD2E | LVD1E | — | EXVCCINP2 | — | — | — |
| Value after reset: | 0 | 0 | 0 *1 | 0 | 0 | 0 | 0 | 0 |

Note 1. The value after a reset is 1 when the OSF1.STUPLVD1REN bit is 0.

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b2 to b0 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |
| b3 | EXVCCINP2 | Voltage Detection 2 Comparison Voltage External Input Select *1 | 0: Power supply voltage (VCC)<br>1: CMPA2 pin input voltage | R/W |
| b4 | — | Reserved | This bit is read as 0. The write value should be 0. | R/W |
| b5 | LVD1E | Voltage Detection 1 Enable | 0: Voltage detection 1 circuit disabled<br>1: Voltage detection 1 circuit enabled | R/W |
| b6 | LVD2E | Voltage Detection 2 Enable | 0: Voltage detection 2 circuit disabled<br>1: Voltage detection 2 circuit enabled | R/W |
| b7 | — | Reserved | This bit is read as 0. The write value should be 0. | R/W |

Note: •Set the PRCR.PRC3 bit to 1 (write enabled) before rewriting this register.
Note 1. The EXVCCINP2 bit can be changed only when the LVD1E and LVD2E bits are both 0 (voltage detection 1 circuit and voltage detection 2 circuit disabled).

### LVD1E Bit (Voltage Detection 1 Enable)

When using voltage detection 1 interrupt/reset or the LVD1SR.LVD1MON flag, set the LVD1E bit to 1. The voltage detection 1 circuit starts once td(E-A) passes after the LVD1E bit value is changed from 0 to 1.

### LVD2E Bit (Voltage Detection 2 Enable)

When using voltage detection 2 interrupt/reset or the LVD2SR.LVD2MON flag, set the LVD2E bit to 1. The voltage detection 2 circuit starts once td(E-A) passes after the LVD2E bit value is changed from 0 to 1.

## 8.2.6 Voltage Detection Level Select Register (LVDLVLR)

Address(es): 0008 C298h

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | LVD2LVL[1:0] | | LVD1LVL[3:0] | | | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 *1 | 1 *1 | 1 *1 | 1 *1 |

Note 1. The value after a reset is the same as the value of the OFS1.STUPLVD1LVL[3:0] bits when the OSF1.STUPLVD1REN bit is 0.

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b3 to b0 | LVD1LVL[3:0] | Voltage Detection 1 Level Select<br>(Standard voltage during drop in voltage) | b3 b0<br>0 1 0 0: 3.10 V<br>0 1 0 1: 3.00 V<br>0 1 1 0: 2.90 V<br>0 1 1 1: 2.79 V<br>1 0 0 0: 2.68 V<br>1 0 0 1: 2.58 V<br>1 0 1 0: 2.48 V<br>1 0 1 1: 2.06 V<br>1 1 0 0: 1.96 V<br>1 1 0 1: 1.86 V<br>Settings other than those listed above are prohibited. | R/W |
| b5, b4 | LVD2LVL[1:0] | Voltage Detection 2 Level Select<br>(Standard voltage during drop in voltage) | b5 b4<br>0 0: 2.90 V<br>0 1: 2.60 V<br>1 0: 2.00 V<br>1 1: 1.80 V *1 | R/W |
| b7, b6 | — | Reserved | This bit is read as 0. The write value should be 0. | R/W |

Note: •Set the PRCR.PRC3 bit to 1 (write enabled) before rewriting this register.
Note 1. Do not set these bits to 11b when the LVCMPCR.EXVCCINP2 bit is 0 (power supply voltage (VCC)).

When changing the LVDLVLR register, first set the LVCMPCR.LVD1E and LVCMPCR.LVD2E bits to 0 (voltage detection n circuit disabled) (n = 1, 2).
When a setting is made so that the voltage detection level range set by the LVD1LVL register overlaps with the range set by the LVD2LVL register, it cannot be specified which of LVD1 and LVD2 is used for voltage detection. For details on the voltage detection level range, refer to section 36, Electrical Characteristics.

## 8.2.7 Voltage Monitoring 1 Circuit Control Register 0 (LVD1CR0)

Address(es): 0008 C29Ah

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | LVD1RN | LVD1RI | — | — | — | LVD1CMPE | — | LVD1RIE |
| Value after reset: | 1 *1 | 0 *2 | 0 | 0 | X | 0 *2 | 0 | 0 *2 |

x: Undefined

Note 1. The value after a reset is 0 when the OSF1.STUPLVD1REN bit is 0.

Note 2. The value after a reset is 1 when the OSF1.STUPLVD1REN bit is 0.

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b0 | LVD1RIE | Voltage Monitoring 1 Interrupt/Reset Enable | 0: Disabled<br>1: Enabled | R/W |
| b1 | — | Reserved | This bit is read as 0. The write value should be 0. | R/W |
| b2 | LVD1CMPE | Voltage Monitoring 1 Circuit Comparison Result Output Enable | 0: Voltage monitoring 1 circuit comparison results output disabled<br>1: Voltage monitoring 1 circuit comparison results output enabled | R/W |
| b3 | — | Reserved | The read value is undefined. The write value should be 0. | R/W |
| b5, b4 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |
| b6 | LVD1RI | Voltage Monitoring 1 Circuit Mode Select | 0: Voltage monitoring 1 interrupt occurs when the voltage passes Vdet1<br>1: Voltage monitoring 1 reset occurs when the voltage falls below Vdet1 | R/W |
| b7 | LVD1RN | Voltage Monitoring 1 Circuit Reset Negation Select | 0: Negation follows a stabilization time (tLVD1) after VCC > Vdet1 is detected.<br>1: Negation follows a stabilization time (tLVD1) after assertion of the voltage monitoring 1 reset. | R/W |

Note: •Set the PRCR.PRC3 bit to 1 (write enabled) before rewriting this register.

### LVD1RIE Bit (Voltage Monitoring 1 Interrupt/Reset Enable)

The LVD1RIE bit is enabled when the LVCMPCR.LVD1E bit is set to 1 (voltage detection 1 circuit enabled) and the LVD1CMPE bit is set to 1 (voltage monitoring 1 circuit comparison results output enabled).

Ensure that neither a voltage monitoring 1 reset nor a voltage monitoring 1 non-maskable interrupt is generated during programming or erasure of the flash memory.

### LVD1RN Bit (Voltage Monitoring 1 Reset Negate Select)

If the LVD1RN bit is to be set to 1 (negation follows a stabilization time after assertion of the voltage monitoring 1 reset), set the LOCOCR.LCSTP bit to 0 (LOCO is operating). Furthermore, if a transition to software standby mode, the only possible value for the LVD1RN bit is 0 (negation follows a stabilization time after VCC > Vdet1 is detected). Do not set the LVD1RN bit to 1 (negation follows a stabilization time after assertion of the voltage monitoring 1 reset).

## 8.2.8 Voltage Monitoring 2 Circuit Control Register 0 (LVD2CR0)

Address(es): 0008 C29Bh

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | LVD2RN | LVD2RI | — | — | — | LVD2CMPE | — | LVD2RIE |
| Value after reset: | 1 | 0 | 0 | 0 | X | 0 | 0 | 0 |

x: Undefined

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b0 | LVD2RIE | Voltage Monitoring 2 Interrupt/Reset Enable | 0: Disabled<br>1: Enabled | R/W |
| b1 | — | Reserved | This bit is read as 0. The write value should be 0. | R/W |
| b2 | LVD2CMPE | Voltage Monitoring 2 Circuit Comparison Result Output Enable | 0: Voltage monitoring 2 circuit comparison results output disabled<br>1: Voltage monitoring 2 circuit comparison results output enabled | R/W |
| b3 | — | Reserved | The read value is undefined. The write value should be 0. | R/W |
| b5, b4 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |
| b6 | LVD2RI | Voltage Monitoring 2 Circuit Mode Select | 0: Voltage monitoring 2 interrupt during Vdet2 passage<br>1: Voltage monitoring 2 reset enabled when the voltage falls to and below Vdet2 | R/W |
| b7 | LVD2RN | Voltage Monitoring 2 Circuit Reset Negation Select | 0: Negation follows a stabilization time (tLVD2) after VCC or the CMPA2 pin > Vdet2 is detected.<br>1: Negation follows a stabilization time (tLVD2) after assertion of the voltage monitoring 2 reset. | R/W |

Note: •Set the PRCR.PRC3 bit to 1 (write enabled) before rewriting this register.

### LVD2RIE Bit (Voltage Monitoring 2 Interrupt/Reset Enable)

The LVD2RIE bit is enabled when the LVCMPCR.LVD2E bit is set to 1 (voltage detection 2 circuit enabled) and the LVD2CMPE bit is set to 1 (voltage monitoring 2 circuit comparison results output enabled).
Ensure that neither a voltage monitoring 2 reset nor a voltage monitoring 2 non-maskable interrupt is generated during programming or erasure of the flash memory.

### LVD2RN Bit (Voltage Monitoring 2 Reset Negate Select)

If the LVD2RN bit is to be set to 1 (negation follows a stabilization time after assertion of the voltage monitoring 2 reset), set the LOCOCR.LCSTP bit to 0 (LOCO is operating). Furthermore, if a transition to software standby mode, the only possible value for the LVD2RN bit is 0 (negation follows a stabilization time after VCC or the CMPA2 pin > Vdet2 is detected). Do not set the LVD2RN bit to 1 (negation follows a stabilization time after assertion of the voltage monitoring 2 reset).

## 8.3 VCC Input Voltage Monitor

### 8.3.1 Monitoring Vdet1

After making the following settings, the LVD1SR.LVD1MON flag can be used to monitor the results of comparison by voltage monitor 1.

(1) Specify the detection voltage by setting the LVDLVLR.LVD1LVL[3:0] bits (voltage detection 1 level select).
(2) Set the LVCMPCR.LVD1E bit to 1 (voltage detection 1 circuit enabled).
(3) After waiting for td(E-A), set the LVD1CR0.LVD1CMPE bit to 1 (voltage monitoring 1 circuit comparison results output enabled).

### 8.3.2 Monitoring Vdet2

After making the following settings, the LVD2SR.LVD2MON flag can be used to monitor the results of comparison by voltage monitor 2.

(1) Specify the detection voltage by setting the LVDLVLR.LVD2LVL[3:0] bits (voltage detection 2 level select).
(2) Clear the LVCMPCR.EXVCCINP2 bit to 0 (VCC voltage) or set it to 1 (CMPA2 pin input voltage).
(3) Set the LVCMPCR.LVD2E bit to 1 (voltage detection 2 circuit enabled).
(4) After waiting for td(E-A), set the LVD2CR0.LVD2CMPE bit to 1 (voltage monitoring 2 circuit comparison results output enabled).

## 8.4 Interrupt and Reset from Voltage Monitoring 1

Table 8.3 shows the procedures for setting bits related to the voltage monitoring 1 interrupt and voltage monitoring 1 reset. Table 8.4 shows the procedures for stopping bits related to the voltage monitoring 1 interrupt and voltage monitoring 1 reset. Figure 8.4 shows an example of operations for a voltage monitoring 1 interrupt. For the operation of the voltage monitoring 1 reset, see Figure 6.2 in section 6, Resets.

**Table 8.3 Procedures for Setting Bits Related to the Voltage Monitoring 1 Interrupt and Voltage Monitoring 1 Reset**

| Step | Voltage Monitoring 1 Interrupt, Voltage Monitoring 1 ELC Event Output | Voltage Monitoring 1 Reset |
|---|---|---|
| 1*1 | Select the detection voltage by setting the LVDLVLR.LVD1LVL[3:0] bits. | |
| 2*1 | Clear the LVD1CR0.LVD1RI bit to 0 (voltage monitoring 1 interrupt). | Set the LVD1CR0.LVD1RI bit to 1 (voltage monitoring 1 reset).<br>Select the type of reset negation by setting the LVD1CR0.LVD1RN bit. |
| 3 | Select the timing of interrupt requests by setting the LVD1CR1.LVD1IDTSEL[1:0] bits.<br>Select the type of interrupt by setting the LVD1CR1.LVD1IRQSEL bit. | — |
| 4 | — | Set the LVD1CR0.LVD1RIE bit to 1 (voltage monitoring 1 interrupt/reset enabled). |
| 5*1 | Set the LVCMPCR.LVD1E bit to 1 (voltage detection 1 circuit enabled). | |
| 6*1 | Wait for at least td(E-A). | |
| 7 | Set the LVD1CR0.LVD1CMPE bit to 1 (voltage monitoring 1 circuit comparison results output enabled). | |
| 8 | Clear the LVD1SR.LVD1DET bit to 0. | — |
| 9 | Set the LVD1CR0.LVD1RIE bit to 1 (voltage monitoring 1 interrupt/reset enabled). | — |

Note 1. Steps 1, 2, 5, and 6 are not required if operation is with the setting to select the voltage monitoring 1 interrupt (LVD1CR0.LVD1RI = 0) and operation can be restarted by simply changing the settings of the LVD1CR1.LVD1IRQSEL and LVD1IDTSEL[1:0] bits after monitoring is stopped or if restarting is in a case where the settings related to the voltage-detection circuit were not changed after monitoring was stopped. When changes are to be made and operation is with the setting to select the voltage monitoring 1 reset (LVD1CR0.LVD1RI = 1), proceed through all steps from 1 to 9.

**Table 8.4 Procedures for Stopping Bits Related to the Voltage Monitoring 1 Interrupt and Voltage Monitoring 1 Reset**

| Step | Voltage Monitoring 1 Interrupt, Voltage Monitoring 1 ELC Event Output | Voltage Monitoring 1 Reset |
|---|---|---|
| 1 | Clear the LVD1CR0.LVD1RIE bit to 0 (voltage monitoring 1 interrupt/reset disabled). | — |
| 2 | Clear the LVD1CR0.LVD1CMPE bit to 0 (voltage monitoring 1 circuit comparison results output disabled). | |
| 3*1 | Clear the LVCMPCR.LVD1E bit to 0 (voltage detection 1 circuit disabled). | |
| 4 | — | Clear the LVD1CR0.LVD1RIE bit to 0 (voltage monitoring 1 interrupt/reset disabled). |
| 5 | Modify settings of bits related to the voltage detection circuit registers other than LVCMPCR.LVD1E, LVD1CR0.LVD1RIE, and LVD1CR0.LVD1CMPE. | |

Note 1. Step 3 is not required if operation is with the setting to select the voltage monitoring 1 interrupt (LVD1CR0.LVD1RI = 0) and operation can be restarted by simply changing the settings of the LVD1CR1.LVD1IRQSEL and LVD1IDTSEL[1:0] bits after monitoring is stopped or if restarting is in a case where the settings related to the voltage-detection circuit were not changed after monitoring was stopped. When changes are to be made and operation is with the setting to select the voltage monitoring 1 reset (LVD1CR0.LVD1RI = 1), proceed through all steps from 1 to 5.

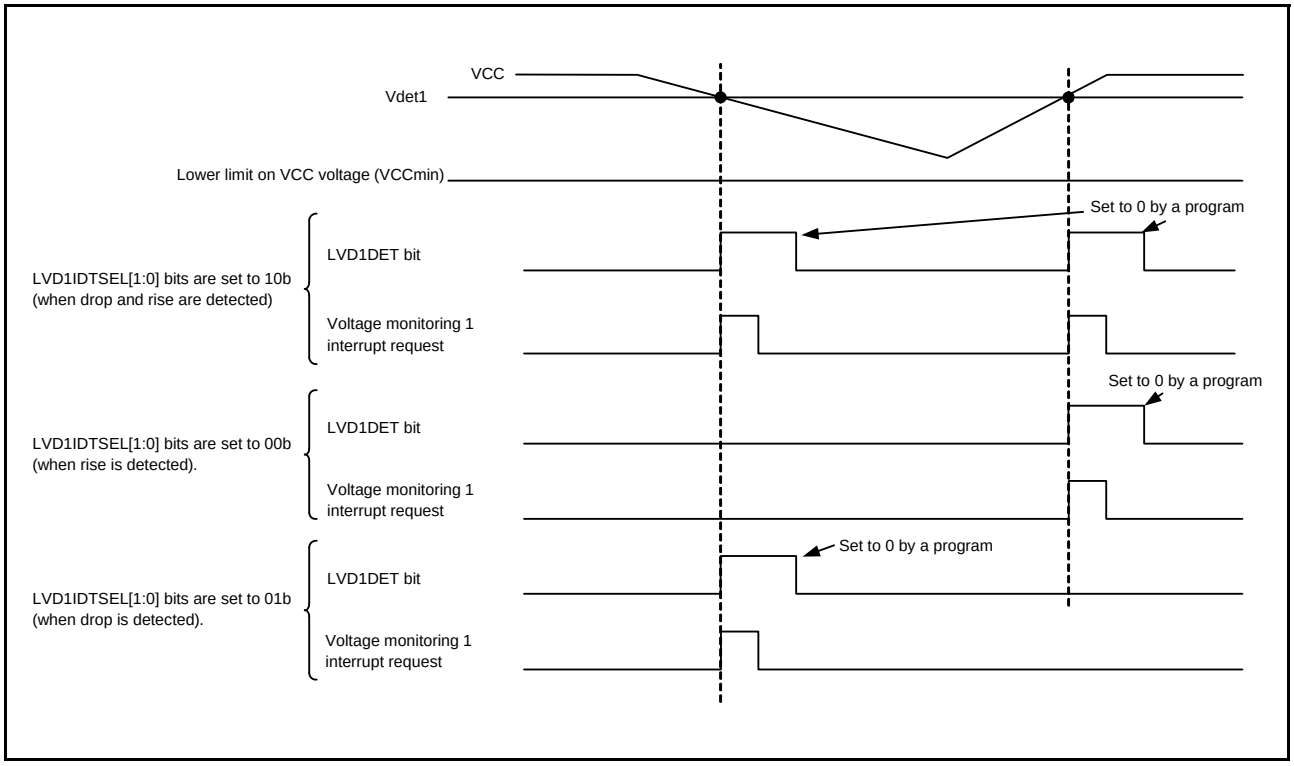



**Figure 8.4 Example of Voltage Monitoring 1 Interrupt Operation**

## 8.5 Interrupt and Reset from Voltage Monitoring 2

Table 8.5 shows the procedures for setting bits related to the voltage monitoring 2 interrupt and voltage monitoring 2 reset. Table 8.6 shows the procedure for stopping bits related to the voltage monitoring 2 interrupt and voltage monitoring 2 reset. Figure 8.5 shows an example of operations for a voltage monitoring 2 interrupt. For the operation of the voltage monitoring 2 reset, see Figure 6.2 in section 6, Resets.

**Table 8.5 Procedures for Setting Bits Related to the Voltage Monitoring 2 Interrupt and Voltage Monitoring 2 Reset**

| Step | Voltage Monitoring 2 Interrupt | Voltage Monitoring 2 Reset |
|---|---|---|
| 1*1 | Select the detection voltage by setting the LVDLVLR.LVD2LVL[1:0] bits. | |
| 2*1 | Set the LVCMPCR.EXVCCINP2 bit to 0 (VCC voltage) or set it to 1 (CMPA2 pin input voltage). | |
| 3*1 | Clear the LVD2CR0.LVD2RI bit (voltage monitoring 2 interrupt). | Set the LVD2CR0.LVD2RI bit to 1 (voltage monitoring 2 reset).<br>Select the type of reset negation by setting the LVD2CR0.LVD2RN bit. |
| 4 | Select the timing of interrupt requests by setting the LVD2CR1.LVD2IDTSEL[1:0] bits.<br>Select the type of interrupt by setting the LVD2CR1.LVD2IRQSEL bit. | — |
| 5 | — | Set the LVD2CR0.LVD2RIE bit to 1 (voltage monitoring 2 interrupt/reset enabled). |
| 6*1 | Set the LVCMPCR.LVD2E bit to 1 (voltage detection 2 circuit enabled). | |
| 7*1 | Wait for at least td(E-A). | |
| 8 | Set the LVD2CR0.LVD2CMPE bit to 1 (voltage monitoring 2 circuit comparison results output enabled). | |
| 9 | Set the LVD2SR.LVD2DET bit to 0. | — |
| 10 | Set the LVD2CR0.LVD2RIE bit to 1 (voltage monitoring 2 interrupt/reset enabled) | — |

Note 1. Steps 1, 2, 3, 6, and 7 are not required if operation is with the setting to select the voltage monitoring 2 interrupt (LVD2CR0.LVD2RI = 0) and operation can be restarted by simply changing the settings of the LVD2CR1.LVD2IRQSEL and LVD2IDTSEL[1:0] bits after monitoring is stopped or if restarting is in a case where the settings related to the voltage-detection circuit were not changed after monitoring was stopped. When changes are to be made and operation is with the setting to select the voltage monitoring 2 reset (LVD2CR0.LVD2RI = 1), proceed through all steps from 1 to 10.

**Table 8.6 Procedures for Stopping Bits Related to the Voltage Monitoring 2 Interrupt and Voltage Monitoring 2 Reset**

| Step | Voltage Monitoring 2 Interrupt | Voltage Monitoring 2 Reset |
|---|---|---|
| 1 | Clear the LVD2CR0.LVD2RIE bit to 0 (voltage monitoring 2 interrupt/reset disabled). | — |
| 2 | Clear the LVD2CR0.LVD2CMPE bit to 0 (voltage monitoring 2 circuit comparison results output disabled). | |
| 3*1 | Clear the LVCMPCR.LVD2E bit to 0 (voltage monitoring 2 circuit disabled). | |
| 4 | — | Clear the LVD2CR0.LVD2RIE bit to 0 (voltage monitoring 2 interrupt/reset disabled). |
| 5 | Modify settings of bits related to the voltage detection circuit registers other than LVCMPCR.LVD2E, LVD2CR0.LVD2RIE, and LVD2CR0.LVD2CMPE. | |

Note 1. Step 3 is not required if operation is with the setting to select the voltage monitoring 2 interrupt (LVD2CR0.LVD2RI = 0) and operation can be restarted by simply changing the settings of the LVD2CR1.LVD2IRQSEL and LVD2IDTSEL[1:0] bits after monitoring is stopped or if restarting is in a case where the settings related to the voltage-detection circuit were not changed after monitoring was stopped. When changes are to be made and operation is with the setting to select the voltage monitoring 2 reset (LVD2CR0.LVD2RI = 1), proceed through all steps from 1 to 5.

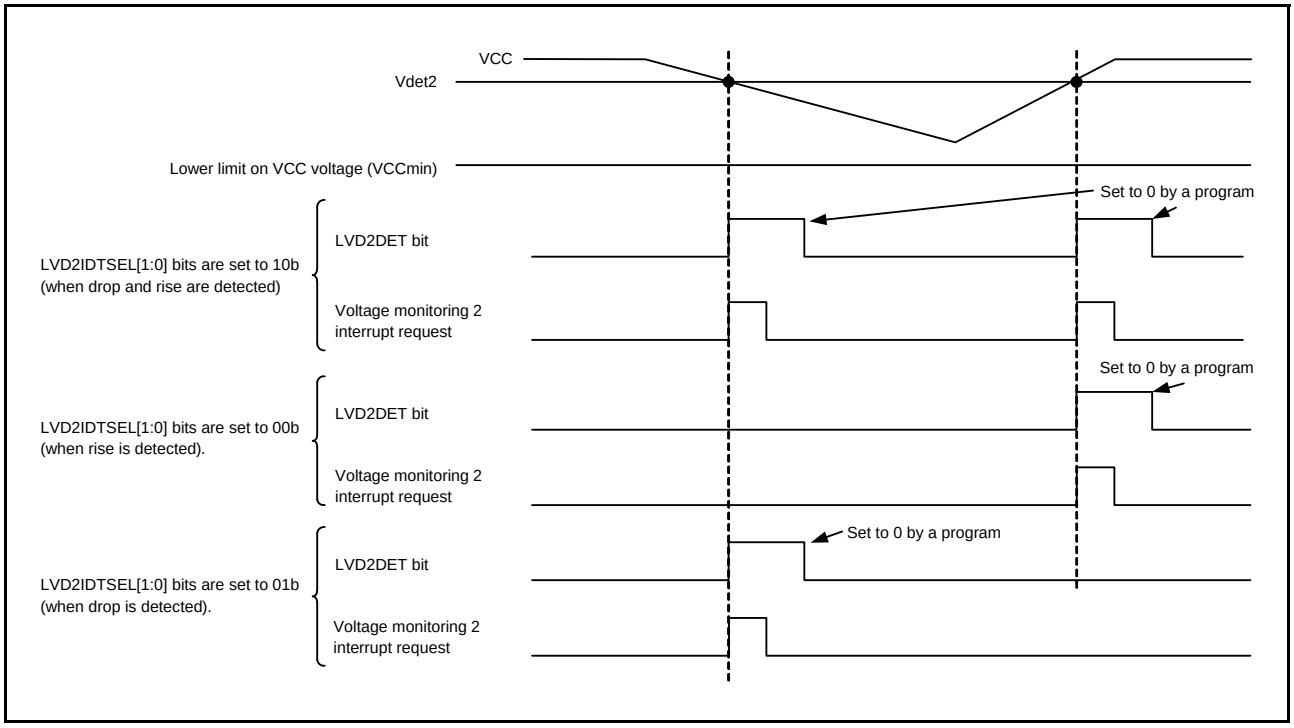



**Figure 8.5 Example of Voltage Monitoring 2 Interrupt Operation**

## 8.6 Event Link Output

The LVD can output the event signals to the event link controller (ELC).

(1) Vdet1 passage detection event output

The LVD outputs the event signal when it is detected that the voltage has passed the Vdet1 voltage while both the voltage detection 1 circuit and the voltage monitoring 1 circuit comparison result output are enabled.

When enabling the LVD's event link output function, be sure to make settings for enabling the LVD before enabling the LVD event link function of the ELC. To stop the LVD's event link output function, be sure to make settings for stopping the LVD before disabling the LVD event link function of the ELC.

### 8.6.1 Interrupt Handling and Event Linking

The LVD has the bits to separately enable or disable the voltage monitoring 1 and 2 interrupts. When an interrupt source is generated and the interrupt is enabled by the interrupt enable bit, the interrupt request signal is output to the CPU. On the contrary, as soon as an interrupt source is generated, the event link signal is output as the event signal to the other module via the ELC regardless of the state of the interrupt enable bit.

It is possible to output voltage monitoring 1 and 2 interrupts in software standby. The event signals for the ELC, however, are output as follows:

- When the events of passing Vdet1 are detected in software standby mode, no event signals are generated for the ELC because no clock is presented in software standby mode. Since the Vdet1 passage detection flags are preserved, however, when the supply of the clock is resumed after restoring from software standby mode, the event signals for the ELC are output according to the state of the Vdet1 passage detection flags.

# 9. Clock Generation Circuit

## 9.1 Overview

This MCU incorporates a clock generation circuit.

Table 9.1 lists the specifications of the clock generation circuit. Figure 9.1 shows a block diagram of the clock generation circuit.

**Table 9.1 Specifications of Clock Generation Circuit**

| Item | Specification |
|---|---|
| Uses | • Generates the system clock (ICLK) to be supplied to the CPU, DTC, ROM, and RAM.<br>• Generates the peripheral module clocks (PCLKB and PCLKD) to be supplied to peripheral modules.<br>The peripheral module clock used as the operating clock is PCLKD for S12AD and PCLKB for other modules.<br>• Generates the FlashIF clock (FCLK) to be supplied to the FlashIF.<br>• Generates the USB clock (UCLK) to be supplied to the USB.<br>• Generates the CAC clock (CACCLK) to be supplied to the CAC.<br>• Generates the RTC-dedicated sub-clock (RTCSCLK) to be supplied to the RTC.<br>• Generates the IWDT-dedicated low-speed clock (IWDTCLK) to be supplied to the IWDT. |
| Operating frequencies*1 | • ICLK: 32 MHz (max)*2<br>• PCLKB: 32 MHz (max)*2<br>• PCLKD: 32 MHz (max)*2<br>• FCLK: 1 to 32 MHz (for programming and erasing the ROM and E2 DataFlash)<br>32 MHz (max) (for reading from the E2 DataFlash)<br>• UCLK: 48 MHz<br>• CACCLK: Same frequency as each oscillator<br>• RTCSCLK: 32.768 kHz<br>• IWDTCLK: 15 kHz |
| Main clock oscillator | • Resonator frequency: 1 to 20 MHz (VCC ≥ 2.4 V), 1 to 8 MHz (VCC < 2.4 V)<br>• External clock input frequency: 20 MHz (max)<br>• Connectable resonator or additional circuit: ceramic resonator, crystal<br>• Connection pins: EXTAL, XTAL<br>• Oscillation stop detection function:<br>When a main clock oscillation stop is detected, the system clock source is switched to LOCO and MTU output can be forcedly driven to high-impedance. |
| Sub-clock oscillator | • Resonator frequency: 32.768 kHz<br>• Connectable resonator or additional circuit: crystal<br>• Connection pin: XCIN, XCOUT |
| PLL circuit*3 | • Input clock source: Main clock<br>• Input pulse frequency division ratio: Selectable from 1, 2, and 4<br>• Input frequency: 4 to 8 MHz<br>• Frequency multiplication ratio: Selectable from 6 and 8<br>• VCO oscillation frequency: 32 to 48 MHz (VCC ≥ 2.4 V) |
| High-speed on-chip oscillator (HOCO) | Oscillation frequency: 32 MHz |
| Low-speed on-chip oscillator (LOCO) | Oscillation frequency: 4 MHz |
| IWDT-dedicated on-chip oscillator | Oscillation frequency: 15 kHz |

Note 1. The maximum operating frequency in high-speed operating mode. For the maximum operating frequency in the other operating modes, refer to section 11.2.5, Operating Power Control Register (OPCCR).
Note 2. The relationship of frequencies must be set as follows. ICLK: FCLK, PCLKB, and PCLKD = N: 1 (N is an integer)
Note 3. The PLL can be used when the external voltage (VCC) is 2.4 V or above.

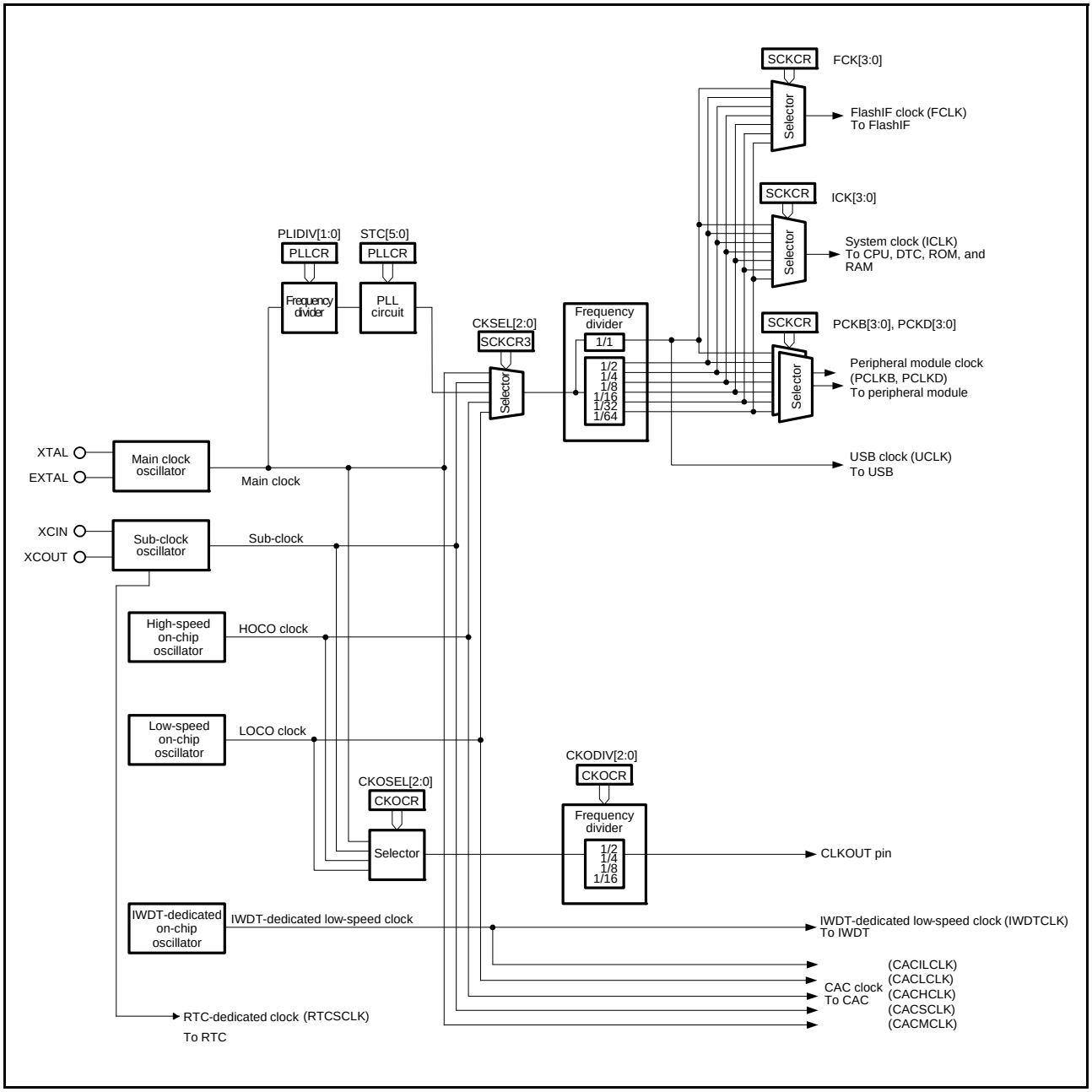



**Figure 9.1 Block Diagram of Clock Generation Circuit**

Table 9.2 lists the I/O pins of the clock generation circuit.

**Table 9.2 I/O Pins of Clock Generation Circuit**

| Pin Name | I/O | Description |
|---|---|---|
| XTAL | Output/Input*1 | These pins are used to connect a crystal. The XTAL pin can also be used to input an external clock. For details, refer to section 9.3.2, External Clock Input. |
| EXTAL | Input | |
| XCIN | Input | These pins are used to connect a 32.768-kHz crystal. |
| XCOUT | Output | |
| CLKOUT | Output | Clock output pin |

Note 1. For external clock input.

## 9.2 Register Descriptions

### 9.2.1 System Clock Control Register (SCKCR)

Address(es): 0008 0020h

| | b31 | b30 | b29 | b28 | b27 | b26 | b25 | b24 | b23 | b22 | b21 | b20 | b19 | b18 | b17 | b16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | FCK[3:0] | | | | ICK[3:0] | | | | — | — | — | — | — | — | — | — |
| Value after reset: | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | PCKB[3:0] | | | | — | — | — | — | PCKD[3:0] | | | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b3 to b0 | PCKD[3:0] | Peripheral Module Clock D (PCLKD) Select*1, *2 | b3 b0<br>0 0 0 0: ×1<br>0 0 0 1: ×1/2<br>0 0 1 0: ×1/4<br>0 0 1 1: ×1/8<br>0 1 0 0: ×1/16<br>0 1 0 1: ×1/32<br>0 1 1 0: ×1/64<br>Settings other than those listed above are prohibited. | R/W |
| b7 to b4 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |
| b11 to b8 | PCKB[3:0] | Peripheral Module Clock B (PCLKB) Select*1, *2 | b11 b8<br>0 0 0 0: ×1<br>0 0 0 1: ×1/2<br>0 0 1 0: ×1/4<br>0 0 1 1: ×1/8<br>0 1 0 0: ×1/16<br>0 1 0 1: ×1/32<br>0 1 1 0: ×1/64<br>Settings other than above are prohibited. | R/W |
| b23 to b12 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |
| b27 to b24 | ICK[3:0] | System Clock (ICLK) Select*1, *2 | b27 b24<br>0 0 0 0: ×1<br>0 0 0 1: ×1/2<br>0 0 1 0: ×1/4<br>0 0 1 1: ×1/8<br>0 1 0 0: ×1/16<br>0 1 0 1: ×1/32<br>0 1 1 0: ×1/64<br>Settings other than above are prohibited. | R/W |
| b31 to b28 | FCK[3:0] | FlashIF Clock (FCLK) Select*1, *2 | b31 b28<br>0 0 0 0: ×1<br>0 0 0 1: ×1/2<br>0 0 1 0: ×1/4<br>0 0 1 1: ×1/8<br>0 1 0 0: ×1/16<br>0 1 0 1: ×1/32<br>0 1 1 0: ×1/64<br>Settings other than above are prohibited. | R/W |

Note: •.Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.
Note 1. The setting for division by one is prohibited if the PLL is selected and the PLL oscillation frequency is 48 MHz.
Note 2. The relationship of frequencies must be set as follows. ICLK: FCLK, PCLKB, and PCLKD = 1: N (N is an integer).

This register cannot be rewritten while the flash memory is being programmed or erased.

When an instruction for writing to SCKCR or SCKCR3 is to follow writing to the SCKCR register, do so in accord with the procedure below.

1. Write to the SCKCR register.
2. Confirm that the value has actually been written to the SCKCR register.
3. Proceed to the next step.

**PCKD[3:0] Bits (Peripheral Module Clock (PCLKD) Select)**

These bits select the frequency of peripheral module clock D (PCLKD).

**PCKB[3:0] Bits (Peripheral Module Clock (PCLKB) Select)**

These bits select the frequency of peripheral module clock B (PCLKB).

**ICK[3:0] Bits (System Clock (ICLK) Select)**

These bits select the frequency of the system clock (ICLK).

**FCK[3:0] Bits (FlashIF Clock (FCLK) Select)**

These bits select the frequency of the FlashIF clock (FCLK).

## 9.2.2 System Clock Control Register 3 (SCKCR3)

Address(es): 0008 0026h

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | CKSEL[2:0] | | | — | — | — | — | — | — | — | — |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b7 to b0 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |
| b10 to b8 | CKSEL[2:0] | Clock Source Select | b10 b8<br>0 0 0: LOCO<br>0 0 1: HOCO<br>0 1 0: Main clock oscillator<br>0 1 1: Sub-clock oscillator<br>1 0 0: PLL circuit<br>Settings other than above are prohibited. | R/W |
| b15 to b11 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |

Note: •Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.

This register cannot be rewritten while the flash memory is being programmed or erased.

**CKSEL[2:0] Bits (Clock Source Select)**

These bits select the source of the system clock (ICLK), peripheral module clock (PCLKB and PCLKD), FlashIF clock (FCLK), and USB clock (UCLK) from low-speed on-chip oscillator (LOCO), high-speed on-chip oscillator (HOCO), the main clock oscillator, the sub-clock oscillator, and the PLL circuit.

Transitions to clock sources which are not in operation are prohibited.

## 9.2.3 PLL Control Register (PLLCR)

Address(es): 0008 0028h

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | STC[5:0] | | | | | | — | — | — | — | — | — | PLIDIV[1:0] | |
| Value after reset: | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b1, b0 | PLIDIV[1:0] | PLL Input Frequency Division Ratio Select | b1 b0<br>0 0: ×1<br>0 1: ×1/2<br>1 0: ×1/4<br>1 1: Setting prohibited | R/W |
| b7 to b2 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |
| b13 to b8 | STC[5:0] | Frequency Multiplication Factor Select | b13 b8<br>0 0 1 0 1 1: ×6<br>0 0 1 1 1 1: ×8<br>Settings other than those listed above are prohibited. | R/W |
| b15, b14 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |

Note: •Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.

Writing to the PLLCR is prohibited when the PLLCR2.PLLEN bit is 0 (PLL is operating).

### PLIDIV[1:0] Bits (PLL Input Frequency Division Ratio Select)

These bits select the frequency division ratio of the PLL clock source.

Set these bits so that the frequency of PLL input signal is within the range of 4 MHz to 8 MHz.

### STC[5:0] Bits (Frequency Multiplication Factor Select)

These bits select the frequency multiplication factor of the PLL circuit.

Set these bits so that the PLL oscillation frequency is within the range of 32 MHz to 48 MHz.

## 9.2.4 PLL Control Register 2 (PLLCR2)

Address(es): 0008 002Ah

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | PLLEN |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b0 | PLLEN | PLL Stop Control | 0: PLL is operating.<br>1: PLL is stopped. | R/W |
| b7 to b1 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |

Note: •Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.

### PLLEN Bit (PLL Stop Control)

This bit runs or stops the PLL circuit.
After setting the PLLEN bit to 0 (PLL is operating), confirm that the OSCOVFSR.PLOVF bit is 1 before switching the system clock to the PLL clock.
That is, a fixed time for stabilization is required after the setting for PLL operation. A fixed time is also required for oscillation to stop after the setting to stop PLL operation. Accordingly, take note of the following limitations when starting and stopping PLL operation.

- After stopping the PLL, confirm that the OSCOVFSR.PLOVF bit is 0 before restarting the PLL.
- Confirm that the PLL is operating and that the OSCOVFSR.PLOVF bit is 1 before stopping the PLL.
- Regardless of whether or not it is selected as the system clock, confirm that the OSCOVFSR.PLOVF bit is 1 before executing a WAIT instruction to place the MCU in software standby mode.
- After stopping the PLL, confirm that the OSCOVFSR.PLOVF bit is 0 and execute a WAIT instruction before entering software standby mode.

When the PLL clock is selected by the SCKCR3.CKSEL[2:0] bits, do not set the PLLEN bit (PLL is stopped) to 1.
When middle-speed operating mode is selected by the OPCCR.OPCM[2:0] bit or low-speed operating mode is selected by the SOPCCR.SOPCM bit, do not set the PLLEN bit to 0 (PLL is operating).

## 9.2.5 Main Clock Oscillator Control Register (MOSCCR)

Address(es): 0008 0032h

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | MOSTP |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b0 | MOSTP | Main Clock Oscillator Stop | 0: Main clock oscillator is operating.<br>1: Main clock oscillator is stopped. | R/W |
| b7 to b1 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |

Note: •Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.

Set this register after setting up the main clock oscillator wait control register.

### MOSTP Bit (Main Clock Oscillator Stop)

This bit runs or stops the main clock oscillator.
After setting the MOSTP bit to 0 (main clock oscillator is operating), read the OSCOVFSR.MOOVF bit to confirm that is has become 1, and then use the main clock.
For the main clock oscillator, a fixed time is required for oscillation to become stable after the settings for operation have been made. Furthermore, a fixed time is required for oscillation to actually stop after the settings to stop oscillation have been made. Accordingly, take note of the following limitations when starting and stopping operation.

- After stopping the main clock oscillator, confirm that the OSCOVFSR.MOOVF bit is 0 before restarting the main clock oscillator.
- Confirm that the main clock oscillator is operating and that the OSCOVFSR.MOOVF bit is 1 before stopping the main clock oscillator.
- Regardless of whether or not it is selected as the system clock, confirm that the OSCOVFSR.MOOVF bit is 1 and execute a WAIT instruction in order to operate the main clock oscillator and place the MCU in software standby mode.
- After stopping the main clock oscillator, confirm that the OSCOVFSR.MOOVF bit is 0 and execute a WAIT instruction before entering software standby mode.

Do not set the MOSTP bit to 1 when one of the following condition is met.

- When the main clock is selected as the clock source for the system clock (SCKCR3.CKSEL[2:0] = 010b)
- When the PLL clock is selected as the clock source for the system clock (SCKCR3.CKSEL[2:0] = 100b)
- When the PLL is operating (PLLCR2.PLLEN = 0)

Do not set the MOSTP bit to 0 when the following condition is met.

- When low-speed operating mode is selected by the SOPCCR.SOPCM bit

## 9.2.6 Sub-Clock Oscillator Control Register (SOSCCR)

Address(es): 0008 0033h

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | SOSTP |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b0 | SOSTP | Sub-Clock Oscillator Stop | 0: Sub-clock oscillator is operating.<br>1: Sub-clock oscillator is stopped. | R/W |
| b7 to b1 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |

Note: •Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.

The XCIN and XCOUT pins are also used as ports. In their initialized state, they function as pins for the sub-clock oscillator.

### SOSTP Bit (Sub-Clock Oscillator Stop)

This bit runs or stops the sub-clock oscillator.
The SOSTP bit and the sub-clock oscillator control bit in RTC control register 3 (RCR3.RTCEN) controls whether to operate or stop the sub-clock oscillator. If one of these bits is set so as to enable the operation, the sub-clock oscillator runs.
When changing the value of the SOSTP bit or RCR3.RTCEN bit, execute subsequent instructions after reading the bit and checking that its value has actually been updated (refer to (2), Notes on writing to I/O registers, in section 5, I/O Registers).
After the setting of the SOSTP bit or the RCR3.RTCEN bit has been changed so that the sub-clock oscillator operates, only start using the sub-clock after the sub-clock oscillation stabilization time ($t_{SUBOSC}$) has elapsed.
That is, a fixed time for stabilization is required after the setting for sub-clock oscillator operation. A fixed time is also required for oscillation to stop after the setting to stop the oscillator. Accordingly, take note of the following limitations when starting and stopping the oscillator.

- When restarting the sub-clock oscillator after it has been stopped, allow at least five cycles of the sub-clock as an interval over which it is still stopped.
- Ensure that oscillation by the sub-clock oscillator is stable when making the setting to stop the sub-clock oscillator.
- Regardless of whether or not it is selected as the system clock, ensure that oscillation by the sub-clock oscillator is stable before executing a WAIT instruction to place the chip on software standby.
- When a transition to software standby mode is to follow the setting to stop the sub-clock oscillator, wait for at least two cycles of the sub-clock oscillator after the setting to stop the sub-clock oscillator and before executing the WAIT instruction.

While the sub-clock oscillator is selected by the SCKCR3.CKSEL[2:0] bits, do not set the SOSTP bit to 1 (sub-clock oscillator is stopped).

## 9.2.7 Low-Speed On-Chip Oscillator Control Register (LOCOCR)

Address(es): 0008 0034h

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | LCSTP |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b0 | LCSTP | LOCO Stop | 0: LOCO is operating.<br>1: LOCO is stopped. | R/W |
| b7 to b1 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |

Note: •Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.

### LCSTP Bit (LOCO Stop)

This bit runs or stops the LOCO.

After the setting of the LCSTP bit has been changed so that the LOCO operates, only start using the LOCO clock after the LOCO clock oscillation stabilization time ($t_{LOCO}$) has elapsed.

That is, a fixed time for stabilization of oscillation is required after the setting for LOCO operation. A fixed time is also required for oscillation to stop after the setting to stop the oscillator. Accordingly, take note of the following limitations when starting and stopping the oscillator.

- When restarting the LOCO after it has been stopped, allow at least five cycles of the LOCO as an interval over which it is still stopped.
- Ensure that oscillation by the LOCO is stable when making the setting to stop the LOCO.
- Regardless of whether or not it is selected as the system clock, ensure that oscillation by the LOCO is stable before executing a WAIT instruction to place the chip on software standby.
- When a transition to software standby mode is to follow the setting to stop the LOCO, wait for at least three cycles of the LOCO after the setting to stop the LOCO and before executing the WAIT instruction.

While the LOCO is selected by the SCKCR3.CKSEL[2:0] bits, do not set the LCSTP bit to 1 (LOCO is stopped).

While low-speed operating mode is selected by the SOPCCR.SOPCM bit, do not set the LCSTP bit to 0 (LOCO is operating).

## 9.2.8 IWDT-Dedicated On-Chip Oscillator Control Register (ILOCOCR)

Address(es): 0008 0035h

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | ILCSTP |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b0 | ILCSTP | IWDT-Dedicated On-Chip Oscillator Stop | 0: IWDT-dedicated on-chip oscillator is operating.<br>1: IWDT-dedicated on-chip oscillator is stopped. | R/W |
| b7 to b1 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |

Note: •Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.

When the IWDT start mode select bit in option function select register 0 (OFS0.IWDTSTRT) is 0 (IWDT is operating), the setting of this register is invalid; it is valid only when the OFS0.IWDTSTRT bit is set to 1 (IWDT is stopped). The ILCSTP bit cannot be changed from 0 (IWDT-dedicated on-chip oscillator is operating) to 1 (IWDT-dedicated on-chip oscillator is stopped) while ILOCOCR is valid.

### ILCSTP Bit (IWDT-Dedicated On-Chip Oscillator Stop)

This bit runs or stops the IWDT-dedicated on-chip oscillator.

After the setting of the ILCSTP bit has been changed so that the IWDT-dedicated on-chip oscillator operates, supply of the clock is started the MCU internally after a fixed time corresponding to the IWDT-dedicated clock oscillation stabilization time ($t_{ILOCO}$) has elapsed.

If the IWDT-dedicated clock is to be used, only start using the oscillator after this wait time has elapsed.

Ensure that oscillation by the IWDT-dedicated on-chip oscillator is stable before executing a WAIT instruction to place the chip on software standby mode.

## 9.2.9 High-Speed On-Chip Oscillator Control Register (HOCOCR)

Address(es): 0008 0036h

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | HCSTP |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0/1*1 |

Note 1. The HCSTP bit value after a reset is 0 when the HOCO oscillation enable bit in option function select register 1 (OFS1.HOCOEN) is 0. The HCSTP bit value after a reset is 1 when the OFS1.HOCOEN bit is 1.

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b0 | HCSTP | HOCO Stop | 0: HOCO is operating.<br>1: HOCO is stopped. | R/W |
| b7 to b1 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |

Note: •Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.

Set the high-speed on-chip wait control register before setting this register.

### HCSTP Bit (HOCO Stop)

This bit runs or stops the HOCO.

When changing the HCSTP bit from 1 to 0 (i.e. changing the HOCO clock from stopped to operating), confirm that the OSCOVFSR.HCOVF bit is 1 before switching the system clock to the HOCO clock.

That is, a fixed time for stabilization of oscillation is required after the setting for HOCO operation. A fixed time is also required for oscillation to stop after the setting to stop the oscillator. Accordingly, take note of the following limitations when starting and stopping the oscillator.

- After stopping the HOCO, confirm that the OSCOVFSR.HCOVF bit is 0 before restarting the HOCO.
- Confirm that the HOCO is operating and that the OSCOVFSR.HCOVF bit is 1 before stopping the HOCO.
- Regardless of whether or not it is selected as the system clock, confirm that the OSCOVFSR.HCOVF bit is 1 before executing a WAIT instruction to place the MCU in software standby mode.
- After stopping the HOCO, confirm that the OSCOVFSR.HCOVF bit is 0 and execute a WAIT instruction before entering software standby mode.

While the HOCO is selected by the SCKCR3.CKSEL[2:0] bits, do not set the HCSTP bit to 1 (HOCO is stopped).

While low-speed operating mode is selected by the SOPCCR.SOPCM bit, do not set the HCSTP bit to 0 (HOCO is operating).

## 9.2.10 Oscillation Stabilization Flag Register (OSCOVFSR)

Address(es): 0008 003Ch

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | HCOVF | PLOVF | — | MOOVF |
| Value after reset: | 0 | 0 | 0 | 0 | 0/1*1 | 0 | 0 | 0 |

Note 1. The HCOVF value after a reset is 1 when the HOCO oscillation enable bit in option function selection register 1 (OFS1.HOCOEN) is 0. The HCSTP value after a reset is 1 when the OFS1.HOCOEN bit is 0.

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b0 | MOOVF | Main Clock Oscillation Stabilization Flag | 0: Main clock is stopped<br>1: Oscillation is stable and the clock can be used as the system clock*1 | R |
| b1 | — | Reserved | This bit is read as 0. The write value should be 0. | R/W |
| b2 | PLOVF | PLL Clock Oscillation Stabilization Flag | 0: PLL is stopped or not stabilized<br>1: Oscillation is stable and the clock can be used as the system clock | R |
| b3 | HCOVF | HOCO Clock Oscillation Stabilization Flag | 0: HOCO is stopped or not stabilized<br>1: Oscillation is stable and the clock can be used as the system clock*1 | R |
| b7 to b4 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |

Note 1. When an appropriate value is set in the wait control register for each oscillator. If a set value (wait time) is not adequate, clock supply starts before oscillation becomes stable.

The OSCOVFSR register monitors whether oscillation of each oscillator has become stable.
If a wait control register is provided for each oscillator, specify a wait time that is longer than or equal to the stabilization time of the corresponding oscillation circuit.

### MOOVF Flag (Main Clock Oscillator Wait Counter Overflow Flag)

This flag indicates whether oscillation of the main clock is stable.
[Setting condition]

- After the MOSCCR.MOSTP bit is set to 0 (main clock oscillator is operating) when the MOSTP bit is 1 (main clock oscillator is stopped), the corresponding time set in the MOSCWTCR register has elapsed and supply of the main clock is started to the MCU internally.

[Clearing condition]

- After the MOSCCR.MOSTP bit is set to 1, the processing to stop the oscillation of the main clock oscillator is completed.

### PLOVF Flag (PLL Wait Counter Overflow Flag)

This flag indicates whether oscillation of the PLL clock is stable.
[Setting condition]
After the PLLCR.PLLEN is set to 0 (PLL is operating) when the PLLEN bit is 1 (PLL is stopped), the MOOVF flag becomes 1, the PLL clock oscillation stabilization time (tPLL) has elapsed, and supply of the PLL clock is started to the MCU internally.
[Clearing condition]
After the PLLCR.PLLEN bit is set to 1, the processing to stop the oscillation of the PLL is completed.

**HCOVF Flag (HOCO Wait Counter Overflow Flag)**

This flag indicates whether oscillation of the HOCO clock is stable.

[Setting condition]

- After the HOCOCR.HCSTP bit is set to 0 (HOCO is operating) when the HCSTP bit is 1 (HOCO is stopped), the corresponding time set in the register has elapsed and supply of the HOCO clock is started to the MCU internally.

[Clearing condition]

- After the HOCOCR.HCSTP bit is set to 1, the processing to stop the oscillation of the HOCO is completed.

## 9.2.11 Oscillation Stop Detection Control Register (OSTDCR)

Address(es): 0008 0040h

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | OSTDE | — | — | — | — | — | — | OSTDIE |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b0 | OSTDIE | Oscillation Stop Detection Interrupt Enable | 0: The oscillation stop detection interrupt is disabled. Oscillation stop detection is not notified to the POE.<br>1: The oscillation stop detection interrupt is enabled. Oscillation stop detection is notified to the POE. | R/W |
| b6 to b1 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |
| b7 | OSTDE | Oscillation Stop Detection Function Enable | 0: Oscillation stop detection function is disabled.<br>1: Oscillation stop detection function is enabled. | R/W |

Note: •Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.

### OSTDIE Bit (Oscillation Stop Detection Interrupt Enable)

If the oscillation stop detection flag in the oscillation stop detection status register (OSTDSR.OSTDF) requires clearing, do this after clearing the OSTDIE bit to 0. Wait for at least two cycles of PCLKB before again setting the OSTDIE bit to 1. According to the number of cycles for access to read a given I/O register, wait time longer than two cycles of PCLKB may have to be secured.

### OSTDE Bit (Oscillation Stop Detection Function Enable)

This bit enables or disables the oscillation stop detection function.

When the OSTDE bit is 1 (oscillation stop detection function enabled), the LOCO stop bit (LOCOCR.LCSTP) is cleared to 0 and the LOCO operation is started. The LOCO cannot be stopped while the oscillation stop detection function is enabled; writing 1 (LOCO is stopped) to the LOCOCR.LCSTP bit is invalid.

When the oscillation stop detection flag in the oscillation stop detection status register (OSTDSR.OSTDF) is 1 (main clock oscillation stop has been detected), writing 0 to the OSTDE bit is invalid.

When the OSTDE bit is 1, a transition cannot be made to software standby mode. To make a transition to software standby mode, execute the WAIT instruction with the OSTDE bit being 0.

## 9.2.12 Oscillation Stop Detection Status Register (OSTDSR)

Address(es): 0008 0041h

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | OSTDF |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b0 | OSTDF | Oscillation Stop Detection Flag | 0: The main clock oscillation stop has not been detected.<br>1: The main clock oscillation stop has been detected. | R/(W) *1 |
| b7 to b1 | — | Reserved | These bits are read as 0 and cannot be modified. | R |

Note: •Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.
Note 1. This bit can only be set to 0.

### OSTDF Flag (Oscillation Stop Detection Flag)

This bit is a flag to indicate the main clock status. When the OSTDF flag is 1, it indicates that the main clock oscillation stop has been detected.
Once the main clock oscillation stop is detected, the OSTDF flag is not cleared to 0 even though the main clock oscillation is restarted. The OSTDF flag is cleared to 0 by reading 1 from the bit and then writing 0. At least three ICLK cycles of wait time is necessary between writing 0 to the OSTDF flag and reading the OSTDF flag as 0. If the OSTDF flag is cleared to 0 while the main clock oscillation is stopped, the OSTDF flag becomes 0 and then returns to 1.
When the main clock oscillator (010b) or PLL (100b) is selected by the clock source select bits in system clock control register 3 (SCKCR3.CKSEL[2:0]), the OSTDF flag cannot be modified to 0. The OSTDF flag should be set to 0 after switching the clock source to a source other than the main clock oscillator and the PLL.

[Setting condition]

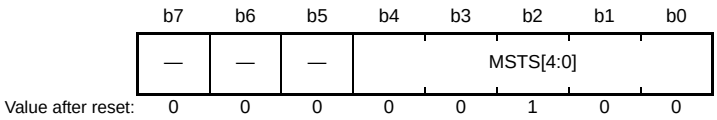
- The main clock oscillation is stopped with the OSTDCR.OSTDE bit being 1 (oscillation stop detection function enabled).

[Clearing condition]

- 1 is read and then 0 is written when the SCKCR3.CKSEL[2:0] bits are neither 010b nor 100b.

## 9.2.13 Main Clock Oscillator Wait Control Register (MOSCWTCR)

Address(es): 0008 00A2h




| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b4 to b0 | MSTS[4:0] | Main Clock Oscillator Wait Time | b4 b0<br>0 0 0 0 0: Wait time = 2 cycles (0.5 µs)<br>0 0 0 0 1: Wait time = 1024 cycles (256 µs)<br>0 0 0 1 0: Wait time = 2048 cycles (512 µs)<br>0 0 0 1 1: Wait time = 4096 cycles (1.024 ms)<br>0 0 1 0 0: Wait time = 8192 cycles (2.048 ms)<br>0 0 1 0 1: Wait time = 16384 cycles (4.096 ms)<br>0 0 1 1 0: Wait time = 32768 cycles (8.192 ms)<br>0 0 1 1 1: Wait time = 65536 cycles (16.384 ms)<br>Settings other than above are prohibited.<br>Wait time when LOCO = 4.0 MHz (0.25 µs, TYP.) | R/W |
| b7 to b5 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |

Note: •Set the PRCR.PRC1 bit to 1 (write enabled) before rewriting this register.

### MSTS[4:0] Bits (Main Clock Oscillator Wait Time)

Set these bits to select the oscillation stabilization wait time of the main clock oscillator.

Set the main clock oscillation stabilization time to longer than or equal to the stabilization time recommended by the oscillator manufacturer. When the main clock is externally input, set these bits to 00000b because the oscillation stabilization time is not required.

The wait time set by the MSTS[4:0] bits is counted using the LOCO clock. The LOCO automatically oscillates when necessary, regardless of the value of the LOCOCR.LOSTP bit.

After the set wait time has elapsed, supply of the main clock is started to the MCU internally and the OSCOVFSR.MOOVF flag becomes 1. If the set wait time is short, supply of the main clock is started before oscillation of the clock becomes stable.

Only rewrite the MOSCWTCR register when the MOSCCR.MOSTP bit is 1 and the OSCOVFSR.MOOVF flag is 0. Do not rewrite this register under any other conditions.

## 9.2.14 High-Speed On-Chip Oscillator Wait Control Register (HOCOWTCR)

Address(es): 0008 00A5h

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | — | HSTS[4:0] | | | | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b4 to b0 | HSTS[4:0] | High-Speed On-Chip Oscillator Wait Time | b4 b0<br>0 0 1 0 1: Wait time = 138 cycles (34.5 µs)*1, *2, *4<br>0 0 1 1 0: Wait time = 266 cycles (66.5 µs)*3, *4<br>Settings other than above are prohibited. | R/W |
| b7 to b5 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |

Note: •Set the PRCR.PRC2 bit to 1 (write enabled) before rewriting this register.
Note 1. If this value is set, the HOCO oscillation stabilization time ($t_{HOCO}$) is not secured, so the HOCO frequency accuracy shown in Electrical Characteristics is not guaranteed when supply of the clock starts. When $t_{HOCO}$ has elapsed after oscillation starts, the HOCO frequency accuracy is as shown in Electrical Characteristics.
Note 2. When the OFS1.HOCOEN bit is set to 0, the HOCO oscillation stabilization time ($t_{HOCO}$) is secured by hardware, so the clock with the accuracy of the HOCO frequency ($f_{HOCO}$) shown in Electrical Characteristics is supplied after release from the CPU reset state.
Note 3. When this value is set, the HOCO oscillation stabilization time ($t_{HOCO}$) is secured and the clock with the accuracy of the HOCO frequency ($f_{HOCO}$) shown in Electrical Characteristics is supplied after release from the CPU reset state.
Note 4. Wait time when LOCO = 4.0 MHz (0.25 µs, TYP.)

### HSTS[4:0] Bits (High-Speed On-Chip Oscillator Wait Time)

These bits are used to select the oscillation stabilization wait time of the HOCO when setting HOCO operation (the HOCOCR.HCSTP bit to 0) and when canceling software standby mode.
Supply of the HOCO clock is started to the MCU internally after the number of LOCO cycles set by the HSTS[4:0] bits has been counted. Counting of LOCO cycles proceeds regardless of the setting of the LOCOCR.LOSTP bit and hardware automatically controls running and stopping the LOCO.
The clock is not supplied to the MCU internally until counting is completed.
After counting is completed, supply of the clock is started to the MCU internally and the OSCOVFSR.HCOVF flag is set to 1.

The HOCOWTCR register can be rewritten under the following cases. Otherwise, do not rewrite this register.

- When the HOCOCR.HCSTP bit is set to 0 (operating), and the OSCOVFSR.HCOVF flag is read and confirmed to be 1.
- When the HOCOCR.HCSTP bit is set to 1 (stopped), and the OSCOVFSR.HCOVF flag is read and confirmed to be 0.

## 9.2.15 CLKOUT Output Control Register (CKOCR)

Address(es): 0008 003Eh

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | CKOSTP | CKODIV[2:0] | | | — | CKOSEL[2:0] | | | — | — | — | — | — | — | — | — |
| Value after reset: | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b7 to b0 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |
| b10 to b8 | CKOSEL[2:0] | CLKOUT Output Source Select | b10 b8<br>0 0 0: LOCO clock<br>0 0 1: HOCO clock<br>0 1 0: Main clock oscillator<br>0 1 1: Sub-clock oscillator<br>Settings other than above are prohibited. | R/W |
| b11 | — | Reserved | This bit is read as 0. The write value should be 0. | R/W |
| b14 to b12 | CKODIV[2:0] | CLKOUT Output Division Ratio Select | b14 b2<br>0 0 0: No division<br>0 0 1: ×1/2<br>0 1 0: ×1/4<br>0 1 1: ×1/8<br>1 0 0: ×1/16<br>Settings other than above are prohibited. | R/W |
| b15 | CKOSTP | CLKOUT Output Stop Control | 0: CLKOUT pin output enabled*1<br>1: CLKOUT pin output disabled | R/W |

Note: •Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.
Note 1. It is also necessary to set the pin function control register and port mode register for the corresponding pin.

### CKOSEL[2:0] Bits (CLKOUT Output Source Select)

Set these bits to select the LOCO clock, HOCO clock, main clock, or sub-clock as the source of the clock to be output from the CLKOUT pin.

### CKODIV[2:0] Bits (CLKOUT Output Division Ratio Select)

Set these bits to select the clock division ratio.
Set the CKOSTP bit to 1 when changing the division ratio.
The division ratio of the output clock frequency should be set to no higher than 8 MHz when VCC is 2.7 V or above, and no higher than 4 MHz when VCC is below 2.7 V.
For details on the characteristics of the clock output from the CLKOUT pin, see Table 36.30, Timing of On-Chip Peripheral Modules (1).

### CKOSTP Bit (CLKOUT Output Stop Control)

Set this bit to enable or disable output from the CLKOUT pin.
When this bit is set to 1, the selected clock is output. When this bit is set to 1, a low level is output.
If the CKOSTP bit is rewritten while the clock is still oscillating, a glitch may be generated in the output.

## 9.2.16 Main Clock Oscillator Forced Oscillation Control Register (MOFCR)

Address(es): 0008 C293h

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | MOSEL | MODRV21 | — | — | — | — | — |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b4 to b0 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |
| b5 | MODRV21 | Main Clock Oscillator Drive Capability Switch | VCC ≥ 2.4 V<br>0: 1 MHz to 10 MHz<br>1: 10 MHz to 20 MHz<br>VCC < 2.4 V<br>0: 1 MHz to 8 MHz<br>1: Setting prohibited | R/W |
| b6 | MOSEL | Main Clock Oscillator Switch | 0: Resonator<br>1: External oscillator input | R/W |
| b7 | — | Reserved | This bit is read as 0. The write value should be 0. | R/W |

Note: •Set the PRCR.PRC1 bit to 1 (write enabled) before rewriting this register.

### MODRV21 Bit (Main Clock Oscillator Drive Capability Switch)

These bits select the drive capability of the main clock oscillator.

### MOSEL Bit (Main Clock Oscillator Switch)

This bit selects the oscillation source of the main clock oscillator.

## 9.3 Main Clock Oscillator

There are two ways of supplying the clock signal from the main clock oscillator: connecting an oscillator or the input of an external clock signal.

### 9.3.1 Connecting a Crystal

Figure 9.2 shows an example of connecting a crystal.
A damping resistor (Rd) should be added, if necessary. Since the resistor values vary depending on the resonator and the oscillation drive capability, use values recommended by the resonator manufacturer. If use of an external feedback resistor (Rf) is directed by the resonator manufacturer, insert an Rf between EXTAL and XTAL by following the instruction.
When connecting a resonator to supply the clock, the frequency of the resonator should be in the frequency range of the resonator for the main clock oscillator described in Table 9.1.

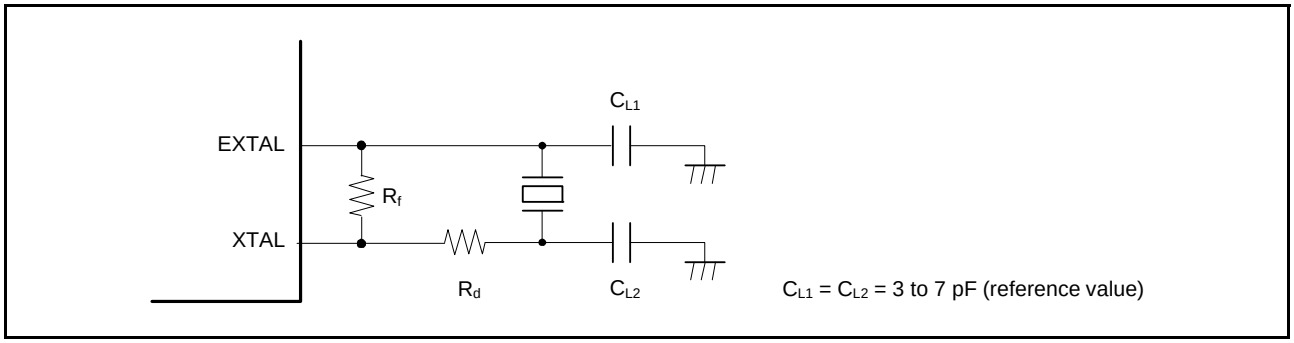



**Figure 9.2 Example of Crystal Connection**

**Table 9.3 Damping Resistance (Reference Values)**

| Frequency (MHz) | 2 | 8 | 16 | 20 |
|---|---|---|---|---|
| Rd (Ω) | 0 | 0 | 0 | 0 |

Figure 9.3 shows an equivalent circuit of the crystal. Use a crystal that has the characteristics shown in Table 9.4 as a reference.

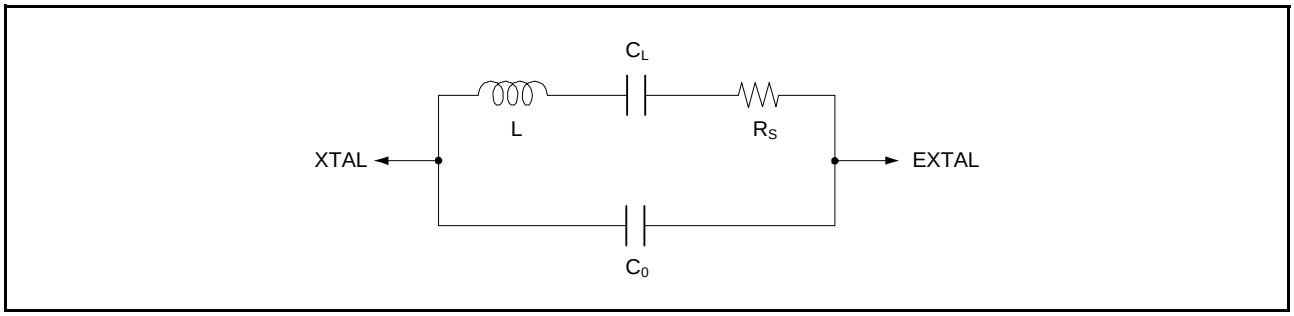



**Figure 9.3 Equivalent Circuit of Crystal**

**Table 9.4 Crystal Characteristics (Reference Values)**

| Frequency (MHz) | 8 | 12 | 16 |
|---|---|---|---|
| $R_S$ max (Ω) | 200 | 120 | 56 |
| $C_0$ max (pF) | 1.3 | 1.3 | 1.4 |

### 9.3.2 External Clock Input

Figure 9.4 shows connection of an external clock. Set the MOFCR.MOSEL bit to 1 if operation is to be driven by an external clock. In this case, the EXTAL pin will be in the Hi-Z state.

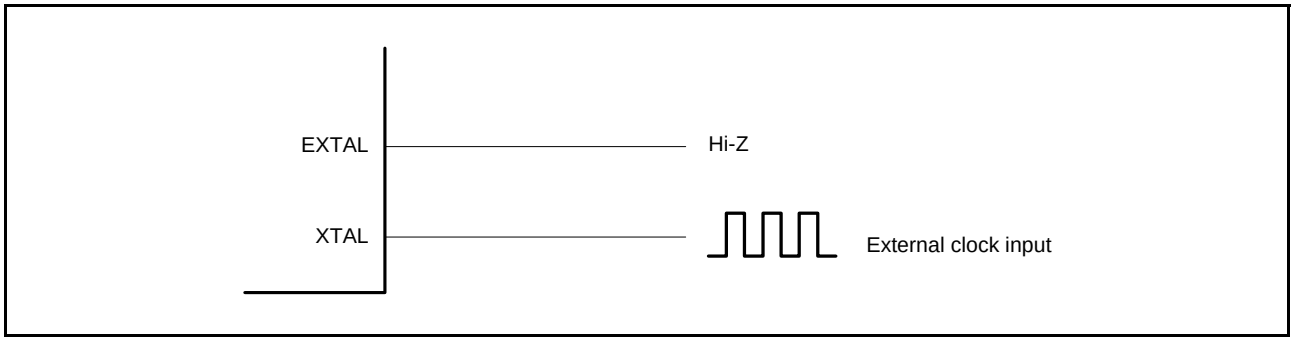



**Figure 9.4 Connection Example of External Clock**

### 9.3.3 Handling of Pins When the Main Clock is Not Used

For details on pin handling when the main clock is not used, refer to section 18.4, Handling of Unused Pins.

### 9.3.4 Notes on the External Clock Input

The frequency of the external clock input can only be changed while the main clock oscillator is stopped. Do not change the frequency of the external clock input while the setting of the main clock oscillator stop bit (MOSCCR.MOSTP) is 0 (main clock oscillator is operating).

## 9.4 Sub-Clock Oscillator

The only way of supplying the clock signal from the sub-clock oscillator is connecting a crystal.

### 9.4.1 Connecting 32.768-kHz Crystal

To supply a clock to the sub-clock oscillator, connect a 32.768-kHz crystal, as shown in Figure 9.5.
A damping resistor Rd should be added, if necessary. Since the resistor values vary depending on the resonator and the oscillation drive capability, use values recommended by the resonator manufacturer. If use of an external feedback resistor (Rf) is directed by the resonator manufacturer, insert an Rf between XCIN and XCOUT by following the instruction. When connecting a resonator to supply the clock, the frequency of the resonator should be in the frequency range of the resonator for the sub-clock oscillator described in Table 9.1.

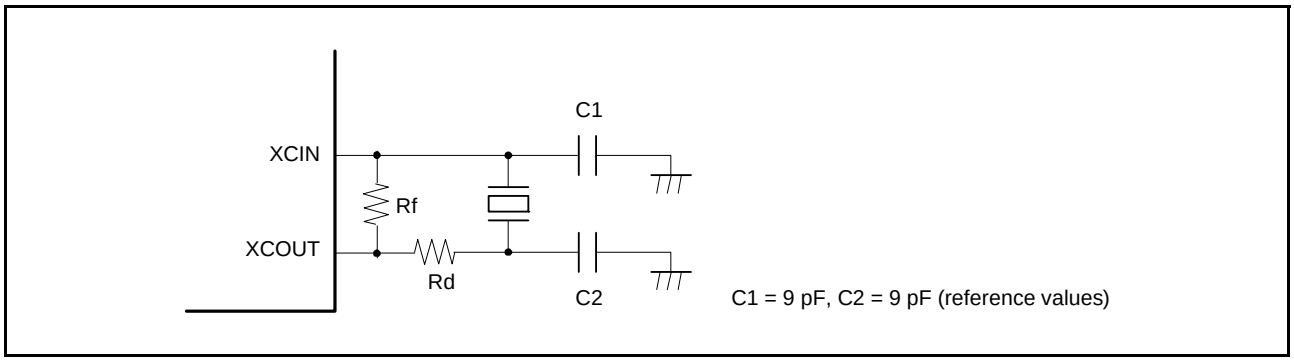



**Figure 9.5 Connection Example of 32.768-kHz Crystal**

Figure 9.6 shows an equivalent circuit for the 32.768-kHz crystal. Use a crystal that has the characteristics listed in Table 9.5.

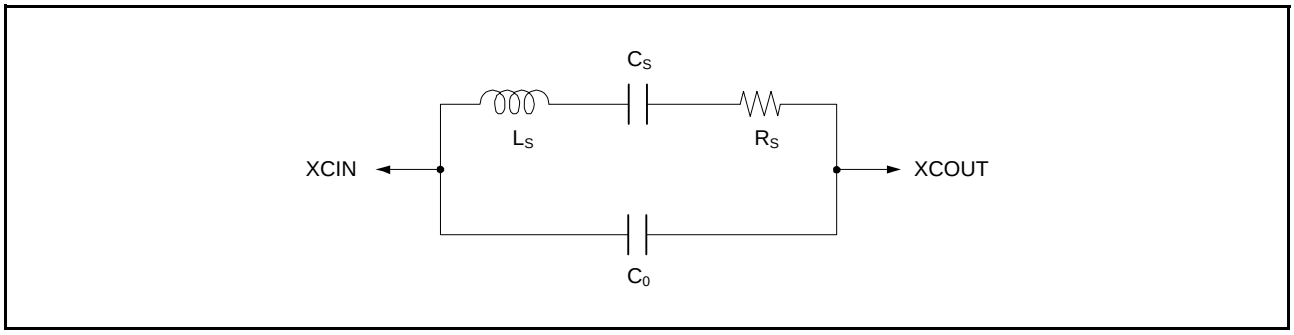



**Figure 9.6 Equivalent Circuit for Crystal**

**Table 9.5 Crystal Characteristics (Reference Values)**

| Frequency (kHz) | 32.768 (Low CL) |
|---|---|
| $R_S$ max (kΩ) | 37 |
| $C_0$ max (pF) | 0.9 |

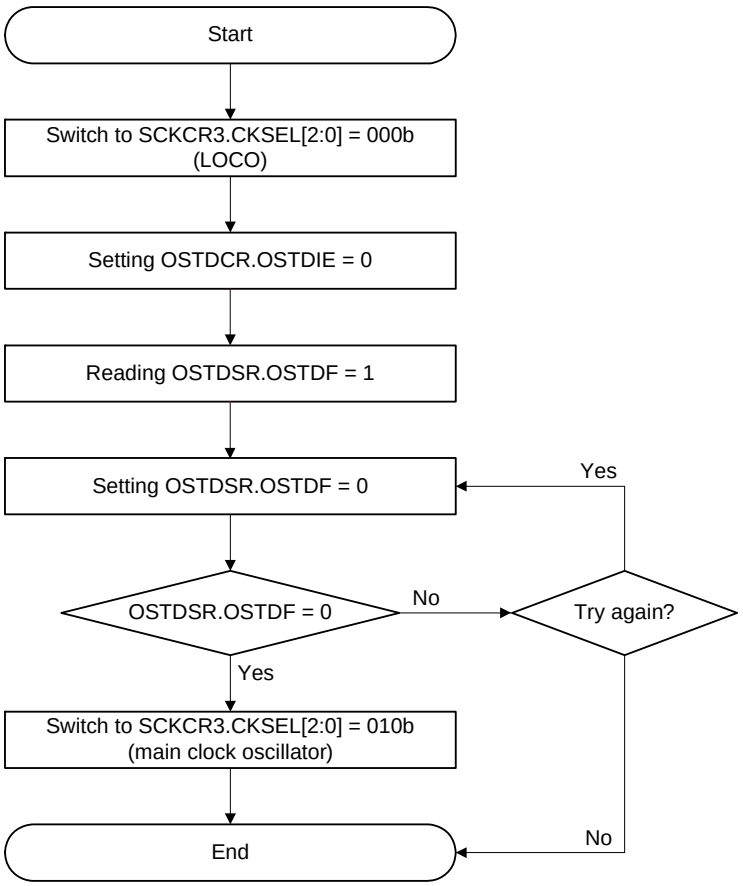
### 9.4.2 Handling of Pins When Sub-Clock is Not Used

When the sub-clock is not used, set the SOSCCR.SOSTP bit to 1 (stopped) and set the RCR3.RTCEN bit to 0 (sub-clock oscillator is stopped) (set general port PH7). When this pin is not also used as port PH7, handle it as an unused pin. For handling of unused pins, refer to section 18.4, Handling of Unused Pins.

## 9.5 Oscillation Stop Detection Function

### 9.5.1 Oscillation Stop Detection and Operation after Detection

The oscillation stop detection function is used to detect the main clock oscillator stop and to supply LOCO clock pulses from the low-speed on-chip oscillator as the system clock source instead of the main clock.
An oscillation stop detection interrupt request can be generated when an oscillation stop is detected. In addition, the MTU output can be forcedly driven to the high-impedance on the detection. For details, refer to section 20, Multi-Function Timer Pulse Unit 2 (MTU2a) and section 21, Port Output Enable 2 (POE2a).
In the MCU, the main clock oscillation stop is detected when the input clock remains to be 0 or 1 for a certain period, for example, due to a malfunction of the main clock oscillator (refer to Figure 36.45, Oscillation Stop Detection Circuit Characteristics, in section 36, Electrical Characteristics).
When an oscillation stop is detected, the main clock selected by the clock source select bits (SCKCR3.CKSEL[2:0]) is switched to the LOCO clock by the corresponding selectors in the former stage. Therefore, if an oscillation stop is detected with the main clock selected as the system clock source, the system clock source is switched to the LOCO clock without a change of CKSEL[2:0].
If an oscillation stop is detected while the PLL clock is selected by the clock source select bits (SCKCR3.CKSEL[2:0]) in system clock control register 3, the SCKCR3.CKSEL[2:0] bit value does not change and the PLL clock remains the system clock source. However, the frequency becomes a free-running oscillation frequency.
Switching between the main clock and LOCO clock is controlled by the oscillation stop detection flag (OSTDSR.OSTDF). The clock source is switched to the LOCO clock when the OSTDF flag is 1, and is switched to the main clock again when the OSTDF flag is cleared to 0. At this time, if the main clock or PLL clock is selected with the CKSEL[2:0] bits, the OSTDF flag cannot be cleared to 0. To switch the clock source to the main clock or PLL clock again after the oscillation stop detection, set the CKSEL[2:0] bits to a clock source other than the main clock or PLL clock and clear the OSTDF flag to 0. After that, check that the OSTDF flag is not 1, and then set the CKSEL[2:0] bits to the main clock or PLL clock after the specified oscillation settling time has elapsed.
After a reset is released, the main clock oscillator is stopped and the oscillation stop detection function is disabled. To enable the oscillation stop detection function, activate the main clock oscillator and write 1 to the oscillation stop detection function enable bit (OSTDCR.OSTDE) after a specified oscillation settling time has elapsed.
The oscillation stop detection function is provided against the main clock stop by an external cause. Therefore, the oscillation stop detection function should be disabled before the main clock oscillator is stopped by the software or a transition is made to software standby mode.
When the main clock or CAC main clock (CACMCLK) is selected as the system clock source, these clocks are switched to the LOCO clock by the oscillation stop detection. The system clock (ICLK) frequency during the LOCO clock operation is specified by the LOCO oscillation frequency and the division ratio set by the system clock (ICLK) select bits (SCKCR.ICK[3:0]).
When the PLL clock is selected as the system clock source, this clock is operated using the PLL free-running oscillation frequency by the oscillation stop detection.




Note: • On return from the oscillation-stopped state, the factor responsible for stopping the main clock oscillation circuit must be removed on the user system to allow the return of oscillation.

**Figure 9.7 Flow of Recovery from Detection of Oscillator Stop**

## 9.5.2 Oscillation Stop Detection Interrupts

An oscillation-stop detection interrupt (OSTDI) will be generated if the oscillation-stop detection flag (OSTDSR.OSTDF) becomes 1 while the oscillation-stop detection interrupt enable bit in the oscillation stop detection control register (OSTDCR.OSTDIE) is 1 (oscillation stop detection interrupt enabled). At this time, the main clock oscillator stop is notified to port output enable 2 (POE). On accepting the notification of the oscillation stop, the POE sets the OSTST high-impedance flag in input level control/status register 3 (ICSR3.OSTSTF) to 1. After the oscillation stop is detected, wait for at lease 10 cycles of PCLK before writing to this ICSR3.OSTSTF flag. When the OSTDSR.OSTDF flag requires clearing, do so after clearing the oscillation stop detection interrupt enable bit in the oscillation stop detection control register (OSTDCR.OSTDIE). Wait for at least two cycles of PCLKB clock before again setting the OSTDCR.OSTDIE bit to 1. According to the number of cycles for access to read a given I/O register, wait time longer than two cycles of PCLKB may have to be secured.

The oscillation stop detection interrupt is a non-maskable interrupt. Since non-maskable interrupts are disabled in the initial state after a reset release, enable the non-maskable interrupts by the software before using oscillation stop detection interrupts. For details, refer to section 14, Interrupt Controller (ICUb).

## 9.6 PLL Circuit

The PLL circuit has a function to multiply the frequency from the oscillator.

## 9.7 Internal Clock

Clock sources of internal clock signals are the main clock, sub-clock, HOCO clock, LOCO clock, and PLL clock, dedicated low-speed clock for the IWDT. The internal clocks listed below are produced from these sources.

(1) Operating clock of the CPU, DTC, ROM, and RAM: System clock (ICLK)
(2) Operating clock of peripheral modules: Peripheral module clock (PCLKB and PCLKD)
(3) Operating clock of the FlashIF: FlashIF clock (FCLK)
(4) Operating clock of USB modules: USB clock (UCLK)
(5) Operating clock for the CAC: CAC clock (CACCLK)
(6) Operating clock for the RTC: RTC-dedicated sub-clock (RTCSCLK)
(7) Operating clock for the IWDT: IWDT-dedicated low-speed clock (IWDTCLK)

Frequencies of the internal clocks are set by the combination of the divisors selected by the SCKCR.FCK[3:0], ICK[3:0], PCKB[3:0], and PCKD[3:0] bits, the clock source selected by the SCKCR3.CKSEL[2:0] bits, and the bits that select the frequency of the PLL circuit (PLLCR.STC[5:0] and PLIDIV[1:0] bits). If the value of any of these bits is changed, subsequent operation will be at the frequency determined by the new value.

### 9.7.1 System Clock

The system clock (ICLK) is used as the operating clock of the CPU, DTC, ROM, and RAM.
The ICLK frequency is specified by the SCKCR.ICK[3:0] bits, and the SCKCR3.CKSEL[2:0] bits, and the PLLCR.STC[5:0] and PLIDIV[1:0] bits.

### 9.7.2 Peripheral Module Clock

The peripheral module clocks (PCLKB and PCLKD) are the operating clocks for use by peripheral modules.
The PCLKB and PCLKD frequencies are specified by the SCKCR.PCKB[3:0] and PCKD[3:0] bits, the SCKCR3.CKSEL[2:0] bits, and the PLLCR.STC[5:0] and PLIDIV[1:0] bits.
The peripheral module clock used as the operating clock is PCLKD for S12AD and PCLKB for other modules.

### 9.7.3 FlashIF Clock

The FlashIF clock (FCLK) is used as the operating clock of the FlashIF.
The FCLK frequency is specified by the SCKCR.FCK[3:0] bits, and the SCKCR3.CKSEL[2:0] bits, and the PLLCR.STC[5:0] and PLIDIV[1:0] bits.

### 9.7.4 USB Clock

The USB clock (UCLK) is an operating clock for the USB module. The UCLK frequency is specified by the SCKCR3.CKSEL[2:0] bits, and the PLLCR.STC[5:0] and PLIDIV[1:0] bits.
A 48-MHz clock must be supplied to the USB module. When the USB module is used, setting must be made so that UCLK is 48 MHz.

### 9.7.5 CAC Clock

The CAC clock (CACCLK) is an operating clock for the CAC module.

The CACCLK clocks include CACMCLK which is generated by the main clock oscillator, CACSCLK which is generated by the sub-clock oscillator, CACHCLK which is generated by the high-speed on-chip oscillator, CACLCLK which is generated by the low-speed on-chip oscillator, and CACILCLK which is generated by the IWDT-dedicated on-chip oscillator.

### 9.7.6 RTC-Dedicated Clock

The RTC-dedicated clock (RTCSCLK) is the operating clock for the RTC.
RTCSCLK is generated by the sub-clock oscillator.

### 9.7.7 IWDT-Dedicated Clock

The IWDT-dedicated clock (IWDTCLK) is the operating clock for the IWDT.
IWDTCLK is internally generated by the IWDT-dedicated on-chip oscillator.

## 9.8 Usage Notes

### 9.8.1 Notes on Clock Generation Circuit

(1) The frequencies of the system clock (ICLK), peripheral module clocks (PCLKB and PCLKD), and FlashIF clock (FCLK) supplied to each module change according to the settings of the SCKCR register. Each frequency should meet the following:
Select each frequency that is within the operation guaranteed range of clock cycle time (tcyc) specified in AC characteristics of electrical characteristics.
The frequencies must not exceed the ranges listed in Table 9.1.
The peripheral modules operate on the PCLKB and PCLKD. Note therefore that the operating speed of modules such as the timer and SCI varies before and after the frequency is changed.

(2) The relationship of frequencies of the system clock (ICLK), peripheral module clocks B and D (PCLKB and PCLKD), and FlashIF clock (FCLK) must be set as follows.
ICLK: FCLK, PCLKB, and PCLKD = N: 1 (N is an integer)

(3) To secure the processing after the clock frequency is changed, modify the pertinent clock control register to change the frequency, and then read the value from the register, and then perform the subsequent processing.

### 9.8.2 Notes on Resonator

Since various resonator characteristics relate closely to the user's board design, adequate evaluation is required on the user side before use, referencing the resonator connection example shown in this section. The circuit constants for the resonator depend on the resonator to be used and the stray capacitance of the mounting circuit. Therefore, the circuit constants should be determined in full consultation with the resonator manufacturer. The voltage to be applied between the resonator pins must be within the absolute maximum rating.

### 9.8.3 Notes on Board Design

When using a crystal, place the resonator and its load capacitors as close to the XTAL and EXTAL pins as possible. Other signal lines should be routed away from the oscillation circuit as shown in Figure 9.8 to prevent electromagnetic induction from interfering with correct oscillation.

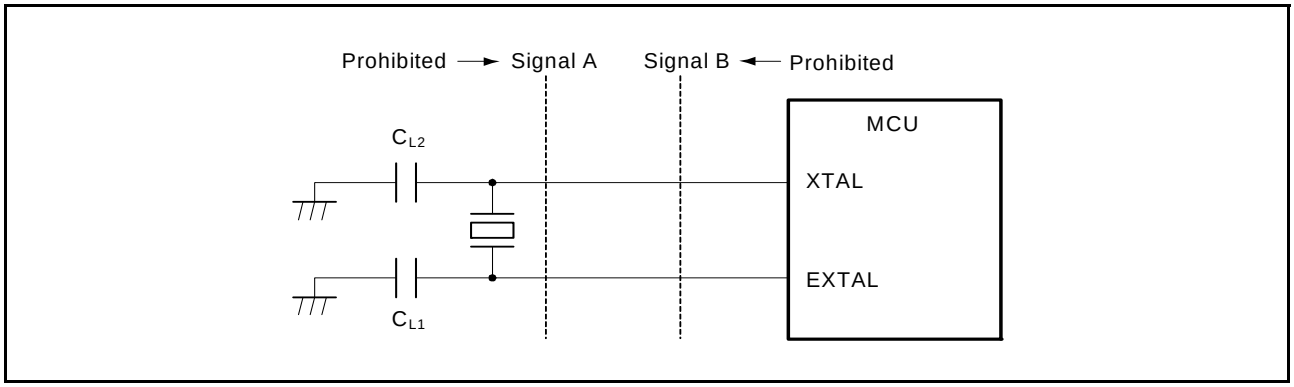
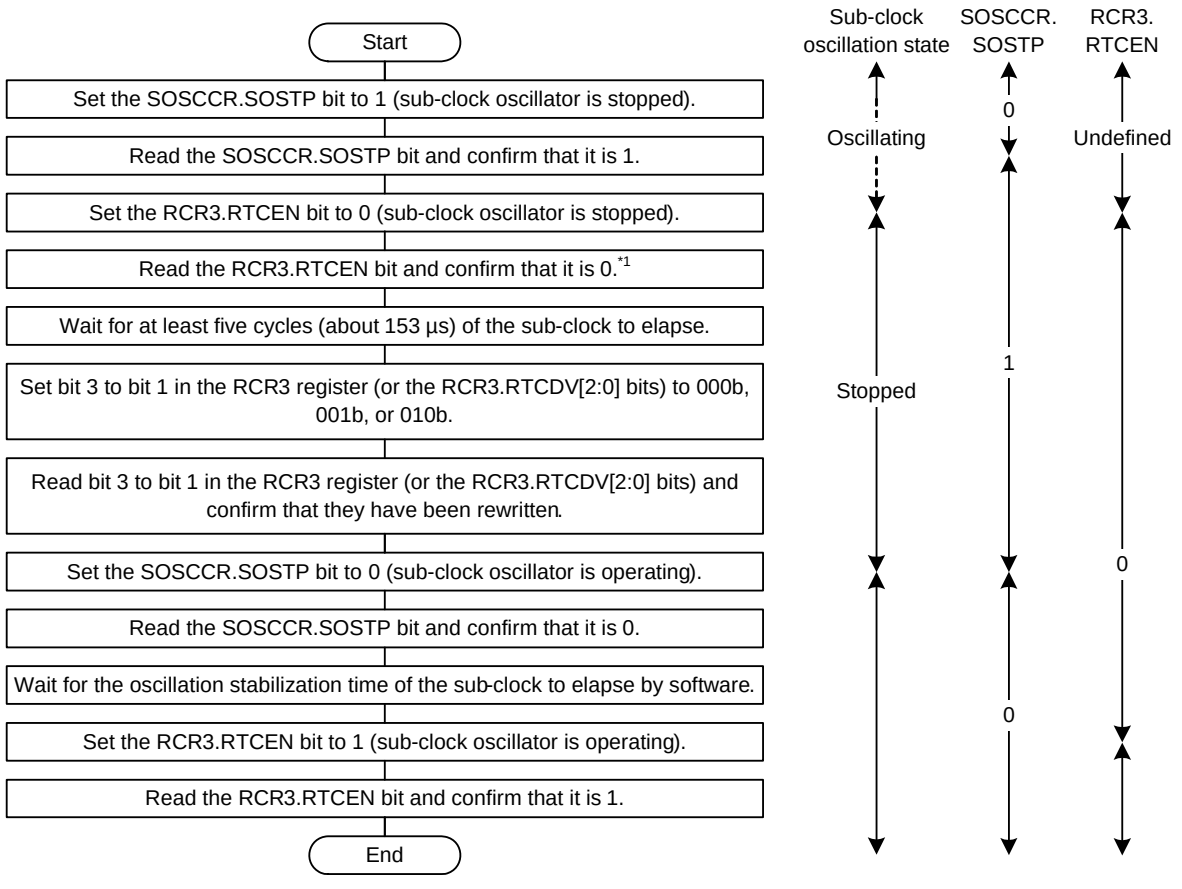



**Figure 9.8 Notes on Board Design for Oscillation Circuit (Applies to the Sub-Clock Oscillator, in Case of the Main Clock Oscillator)**

### 9.8.4 Notes on Sub-Clock

The sub-clock can be used as the system clock, as the count source for the realtime clock, or as both. Take note of the following limitations and points for caution regarding the settings, including when the sub-clock is not in use.

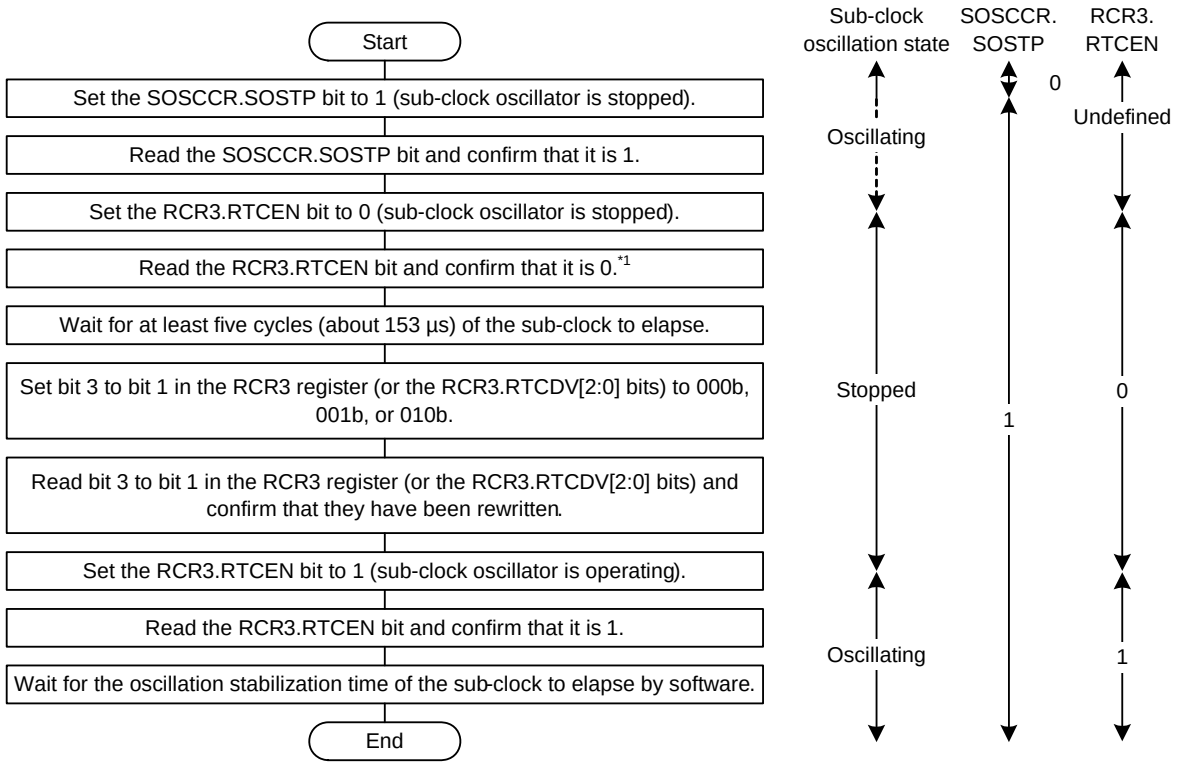
- With regard to making the sub-clock oscillator run or stop, setting either the sub-clock oscillator stop bit in the sub-clock oscillator control register (SOSCCR.SOSTP) or the sub-clock oscillator control bit in RTC control register 3 (RCR3.RTCEN) will make the oscillator run.
- To use the sub-clock as the system clock and as the count source of the realtime clock simultaneously, perform initial settings according to the flowchart example shown in Figure 9.9. After that, perform the clock setting procedure shown in section 23.3.2, Clock and Count Mode Setting Procedure.




Note 1. The values of bit 7 to bit 1 are undefined. Confirm only the value of the RCR3.RTCEN bit.

**Figure 9.9 Example of Initialization Flowchart When Sub-Clock is Used as Count Source of Realtime Clock**

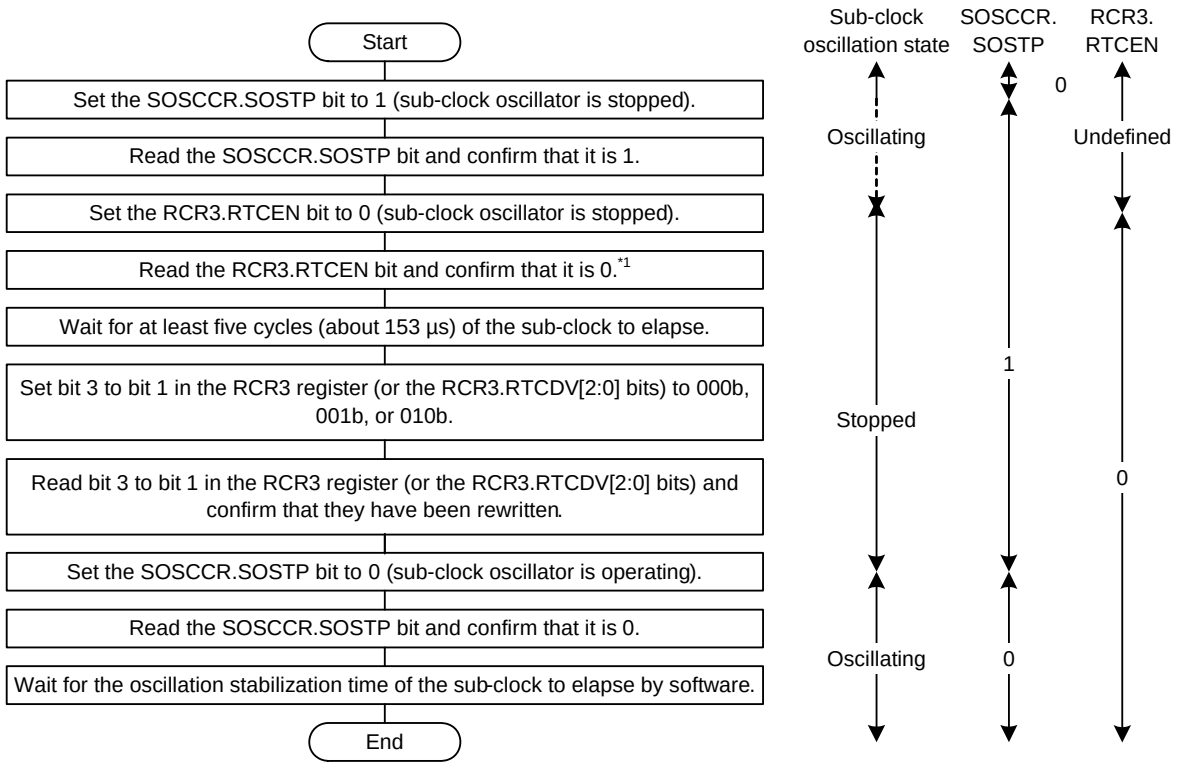
- When using the sub-clock only as the count source of the realtime clock, perform initial settings according to the flowchart example shown in Figure 9.10. After that, perform the clock setting procedure shown in section 23.3.2, Clock and Count Mode Setting Procedure.




Note 1. The values of bit 7 to bit 1 are undefined. Confirm only the value of the RCR3.RTCEN bit.

**Figure 9.10 Example of Initialization Flowchart When Sub-Clock is Used Only as Count Source of Realtime Clock**

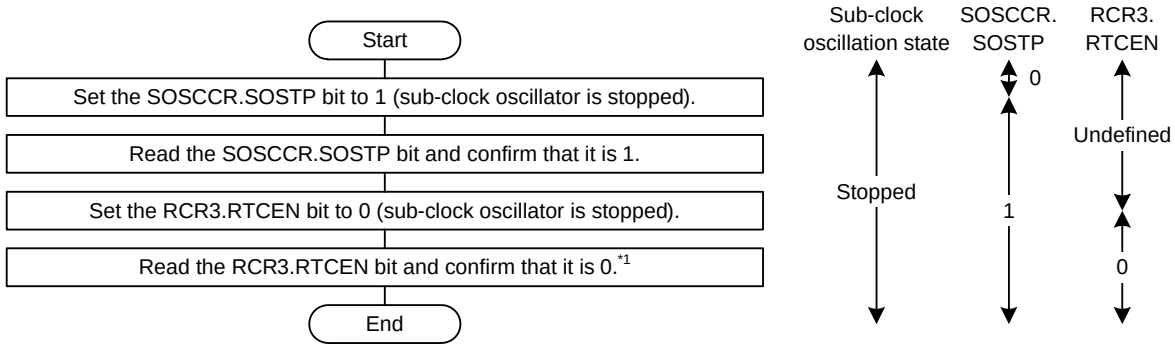
- When using the sub-clock only as the system clock, perform initial settings according to the flowchart example shown in Figure 9.11.




Note 1. The values of bit 7 to bit 1 are undefined. Confirm only the value of the RCR3.RTCEN bit.

**Figure 9.11 Example of Initialization Flowchart When Sub-Clock is Used Only as System Clock**

- When not using the sub-clock, perform initial settings according to the flowchart example in Figure 9.12.




Note 1. The values of bit 7 to bit 1 are undefined. Confirm only the value of the RCR3.RTCEN bit.

**Figure 9.12 Example of Initialization Flowchart When Sub-Clock is Not Used**

- Regardless of the RCR3.RTCEN bit setting, wait until the oscillator stabilization wait time elapses before rewriting the SOSCCR.SOSTP bit to 0 (sub-clock oscillator is operating).
- Since the sub-clock control circuit is in an unstable state after a cold start, it must be initialized regardless of whether or not the sub-clock is in use. The sub-clock is initialized by setting the SOSCCR.SOSTP bit to 1 and the RCR3.RTCEN bit to 0 (sub-clock oscillator is stopped). See section 23.2.19, RTC Control Register 3 (RCR3), for instructions to initialize the RCR3.RTCEN bit.
  Although the sub-clock oscillator pins are not available in 40 or fewer pin package products, initialize the sub-clock control circuit in the same way.
- The RCR3.RTCDV[2:0] bits must also be set when operating the sub-clock oscillator. Set these bits while the sub-clock oscillator is stopped. Do not rewrite these bits while the sub-clock oscillator is operating.
- When successively rewriting the SOSCCR.SOSTP bit followed by the RCR3.RTCEN bit or vice versa, confirm that the first bit rewrite was completed successfully before rewriting the second bit.

# 10. Clock Frequency Accuracy Measurement Circuit (CAC)

The clock frequency accuracy measurement circuit (CAC) monitors the clock frequency based on a reference signal input to the MCU externally or another clock source, and generates interrupts when measurement is completed or the set range is exceeded.

## 10.1 Overview

Table 10.1 shows the specifications of the CAC and Figure 10.1 shows a block diagram of the CAC.

**Table 10.1 CAC Specifications**

| Item | Description |
|---|---|
| Clock frequency measurement | The frequency of the following clocks can be measured.<br>• Clock output from main clock oscillator (main clock)<br>• Clock output from sub-clock oscillator (sub-clock)<br>• Clock output from high-speed on-chip oscillator (HOCO clock)<br>• Clock output from low-speed on-chip oscillator (LOCO clock)<br>• Clock output from IWDT-dedicated on-chip oscillator (IWDTCLK clock)<br>• Peripheral module clock (PCLKB) |
| Selectable function | Digital filter function |
| Interrupt sources | • Measurement end interrupt<br>• Frequency error interrupt<br>• Overflow interrupt |
| Low power consumption function | Module stop state can be set. |

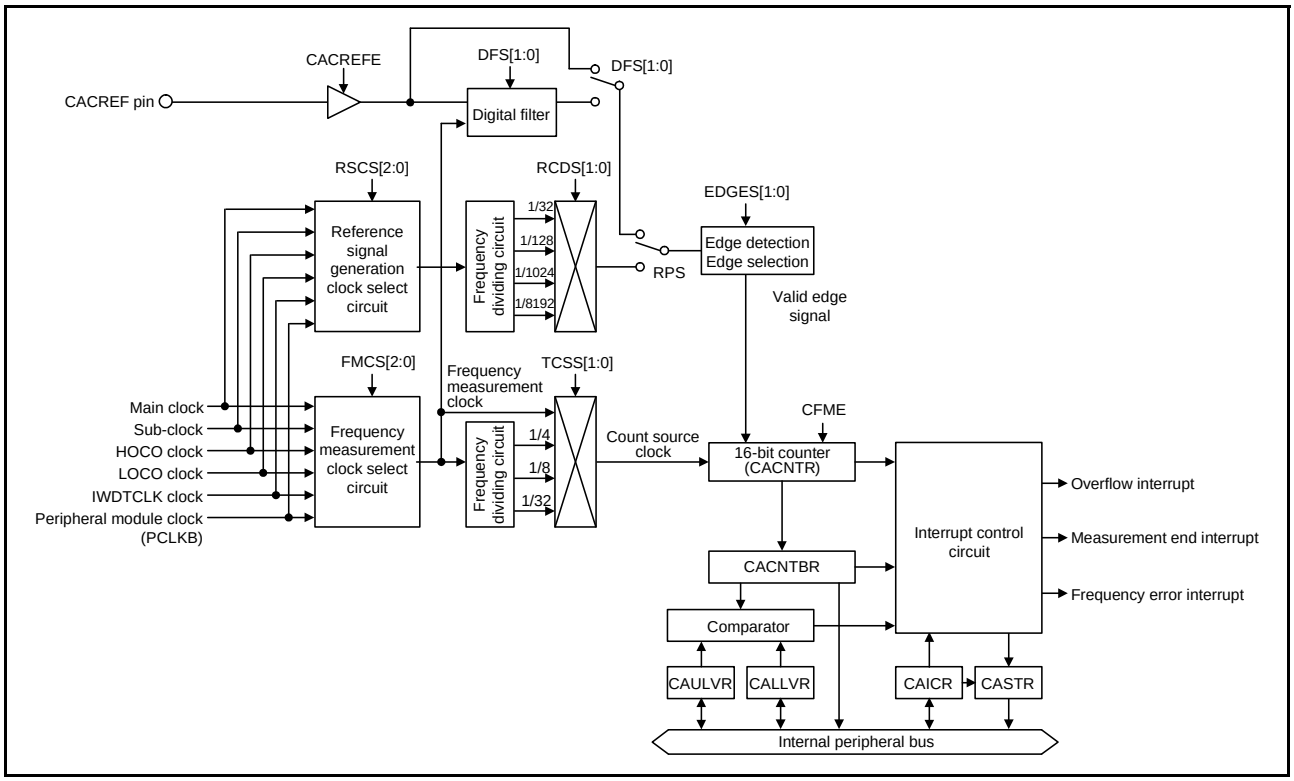



**Figure 10.1 CAC Block Diagram**

Table 10.2 shows the pin configuration of the CAC.

**Table 10.2 Pin Configuration of CAC**

| Pin Name | I/O | Function |
|---|---|---|
| CACREF | Input | Clock frequency accuracy measurement circuit input pin |

## 10.2 Register Descriptions

### 10.2.1 CAC Control Register 0 (CACR0)

Address(es): 0008 B000h

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | CFME |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b0 | CFME | Clock Frequency Measurement Enable | 0: Clock frequency measurement is disabled.<br>1: Clock frequency measurement is enabled. | R/W |
| b7 to b1 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |

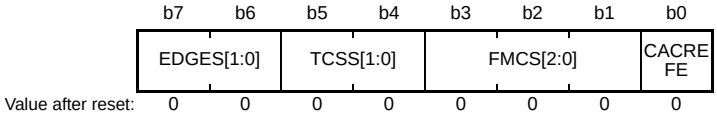
**CFME Bit (Clock Frequency Measurement Enable)**

When this bit is 1, clock frequency measurement is enabled.

When rewriting this bit, more time is required than other bits for the new value to be reflected in the register. Further write access to this bit are ignored until the current write access is reflected in the register. Read the bit to confirm that the rewrite has been reflected in the register.

## 10.2.2 CAC Control Register 1 (CACR1)

Address(es): 0008 B001h




| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b0 | CACREFE | CACREF Pin Input Enable | 0: CACREF pin input is disabled.<br>1: CACREF pin input is enabled. | R/W |
| b3 to b1 | FMCS[2:0] | Frequency Measurement Clock Select | b3 b1<br>0 0 0: Output clock of main clock oscillator<br>0 0 1: Output clock of sub-clock oscillator<br>0 1 0: Output clock of high-speed on-chip oscillator<br>0 1 1: Output clock of low-speed on-chip oscillator<br>1 0 0: Output clock of IWDT-dedicated on-chip oscillator<br>1 0 1: Peripheral module clock (PCLKB)<br>Settings other than above are prohibited. | R/W |
| b5, b4 | TCSS[1:0] | Timer Count Clock Source Select | b5 b4<br>0 0: No division<br>0 1: ×1/4 clock<br>1 0: ×1/8 clock<br>1 1: ×1/32 clock | R/W |
| b7, b6 | EDGES[1:0] | Valid Edge Select | b7 b6<br>0 0: Rising edge<br>0 1: Falling edge<br>1 0: Both rising and falling edges<br>1 1: Setting prohibited | R/W |

Note 1. Set the CACR1 register when the CACR0.CFME bit is 0.

### CACREFE Bit (CACREF Pin Input Enable)

When this bit is 1, the CACREF pin input is enabled.

### FMCS[2:0] Bits (Frequency Measurement Clock Select)

These bits select the clock whose frequency is to be measured.

### TCSS[1:0] Bits (Timer Count Clock Source Select)

These bits select the count clock source for the clock frequency accuracy measurement circuit.

### EDGES[1:0] Bits (Valid Edge Select)

These bits select the valid edge for the reference signal.

## 10.2.3 CAC Control Register 2 (CACR2)

Address(es): 0008 B002h

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | DFS[1:0] | | RCDS[1:0] | | RSCS[2:0] | | | RPS |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b0 | RPS | Reference Signal Select | 0: CACREF pin input<br>1: Internally generated signal | R/W |
| b3 to b1 | RSCS[2:0] | Reference Signal Generation Clock Select | b3 b1<br>0 0 0: Output clock of main clock oscillator<br>0 0 1: Output clock of sub-clock oscillator<br>0 1 0: Output clock of high-speed on-chip oscillator<br>0 1 1: Output clock of low-speed on-chip oscillator<br>1 0 0: Output clock of IWDT-dedicated on-chip oscillator<br>1 0 1: Peripheral module clock (PCLKB)<br>Settings other than above are prohibited. | R/W |
| b5, b4 | RCDS[1:0] | Reference Signal Generation Clock Frequency Division Ratio Select | b5 b4<br>0 0: ×1/32 clock<br>0 1: ×1/128 clock<br>1 0: ×1/1024 clock<br>1 1: ×1/8192 clock | R/W |
| b7, b6 | DFS[1:0] | Digital Filter Selection | b7 b6<br>0 0: Digital filtering is disabled.<br>0 1: The sampling clock for the digital filter is the frequency measuring clock.<br>1 0: The sampling clock for the digital filter is the frequency measuring clock divided by 4.<br>1 1: The sampling clock for the digital filter is the frequency measuring clock divided by 16. | R/W |

Note 1. Set the CACR2 register when the CACR0.CFME bit is 0.

### RPS Bit (Reference Signal Select)

This bit selects whether to use the CACREF pin input or an internally generated signal as the reference signal.

### RSCS[2:0] Bits (Reference Signal Generation Clock Select)

These bits select the clock source for generating the reference signal.

### RCDS[1:0] Bits (Reference Signal Generation Clock Frequency Division Ratio Select)

These bits select the frequency division ratio of the reference signal generation clock.

### DFS[1:0] Bits (Digital Filter Selection)

The setting of these bits enables or disables the digital filter and selects its sampling clock.

## 10.2.4 CAC Interrupt Control Register (CAICR)

Address(es): 0008 B003h

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | OVFFCL | MENDFCL | FERRFCL | — | OVFIE | MENDIE | FERRIE |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b0 | FERRIE | Frequency Error Interrupt Enable | 0: Frequency error interrupt is disabled.<br>1: Frequency error interrupt is enabled. | R/W |
| b1 | MENDIE | Measurement End Interrupt Enable | 0: Measurement end interrupt is disabled.<br>1: Measurement end interrupt is enabled. | R/W |
| b2 | OVFIE | Overflow Interrupt Enable | 0: Overflow interrupt is disabled.<br>1: Overflow interrupt is enabled. | R/W |
| b3 | — | Reserved | This bit is read as 0. The write value should be 0. | R/W |
| b4 | FERRFCL | FERRF Clear | When 1 is written to this bit, the FERRF bit is cleared. This bit is read as 0. | R/W |
| b5 | MENDFCL | MENDF Clear | When 1 is written to this bit, the MENDF bit is cleared. This bit is read as 0. | R/W |
| b6 | OVFFCL | OVFF Clear | When 1 is written to this bit, the OVFF bit is cleared. This bit is read as 0. | R/W |
| b7 | — | Reserved | This bit is read as 0. The write value should be 0. | R/W |

**FERRIE Bit (Frequency Error Interrupt Enable)**

When this bit is 1, the frequency error interrupt is enabled.

**MENDIE Bit (Measurement End Interrupt Enable)**

When this bit is 1, the measurement end interrupt is enabled.

**OVFIE Bit (Overflow Interrupt Enable)**

When this bit is 1, the overflow interrupt is enabled.

**FERRFCL Bit (FERRF Clear)**

Setting this bit to 1 clears the FERRF bit.

**MENDFCL Bit (MENDF Clear)**

Setting this bit to 1 clears the MENDF bit.

**OVFFCL Bit (OVFF Clear)**

Setting this bit to 1 clears the OVFF bit.

## 10.2.5 CAC Status Register (CASTR)

Address(es): 0008 B004h

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | OVFF | MENDF | FERRF |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b0 | FERRF | Frequency Error Flag | [Setting condition]<br>• The clock frequency is outside of the setting range.<br>[Clearing condition]<br>• 1 is written to the FERRFCL bit. | R |
| b1 | MENDF | Measurement End Flag | [Setting condition]<br>• Measurement has finished.<br>[Clearing condition]<br>• 1 is written to the MENDFCL bit. | R |
| b2 | OVFF | Overflow Flag | [Setting condition]<br>• The counter has overflowed.<br>[Clearing condition]<br>• 1 is written to the OVFFCL bit. | R |
| b7 to b3 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |

### FERRF Flag (Frequency Error Flag)

This bit is set to 1 when the clock frequency is outside of the setting range.

### MENDF Flag (Measurement End Flag)

This bit is set to 1 when measurement has finished.

### OVFF Flag (Overflow Flag)

This bit is set to 1 when the counter has overflowed.

## 10.2.6 CAC Upper-Limit Value Setting Register (CAULVR)

Address(es): 0008 B006h

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

CAULVR is a 16-bit readable/writable register that stores the upper-limit value of the frequency.
Write to this register when the CACR0.CFME bit is 0.
The counter value held in CACNTBR can vary with the difference between the phases of the digital filter and edge-detection circuit on the one hand and the signal on the CACREF pin on the other, so ensure that this setting allows an adequate margin.

## 10.2.7 CAC Lower-Limit Value Setting Register (CALLVR)

Address(es): 0008 B008h

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

CALLVR is a 16-bit readable/writable register that stores the lower-limit value of the frequency.
Write to this register when the CACR0.CFME bit is 0.
The counter value held in CACNTBR can vary with the difference between the phases of the digital filter and edge-detection circuit on the one hand and the signal on the CACREF pin on the other, so ensure that this setting allows an adequate margin.

## 10.2.8 CAC Counter Buffer Register (CACNTBR)

Address(es): 0008 B00Ah

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

CACNTBR is a 16-bit read-only register that retains the counter value at the time a valid reference signal edge is input.

## 10.3 Operation

### 10.3.1 Measuring Clock Frequency Based on CACREF Pin Input

Figure 10.2 shows an operating example of the clock frequency accuracy measurement circuit based on the CACREF pin input.
The clock frequency accuracy measurement circuit operates as shown below when measuring the clock frequency.

(1) When the CACR2.RPS bit is 0 and the CACR1.CACREFE bit is 1, if the CACR0.CFME bit is set to 1, the clock frequency measurement based on the CACREF pin input becomes enabled.
(2) After 1 is written to the CFME bit, the timer starts up-counting when the valid edge selected by the EDGES[1:0] bits in CACR1 is input from the CACREF pin.
(3) When the next valid edge is input, the counter value is retained in CACNTBR and compared with the values of CAULVR and CALLVR. If both CACNTBR ≤ CAULVR and CACNTBR ≥ CALLVR are satisfied, only the MENDF flag in CASTR is set to 1 because the clock frequency is correct. If the MENDIE bit in CAICR is 1, a measurement end interrupt is generated.
(4) When the next valid edge is input, the counter value is retained in CACNTBR and compared with the values of CAULVR and CALLVR. In the case of CACNTBR > CAULVR, the FERRF flag in CASTR is set to 1 because the clock frequency is erroneous. If the FERRIE bit in CAICR is 1, a frequency error interrupt is generated. Also, the MENDF flag in CASTR is set to 1. If the MENDIE bit in CAICR is 1, a measurement end interrupt is generated.
(5) When the next valid edge is input, the counter value is retained in CACNTBR and compared with the values of CAULVR and CALLVR. In the case of CACNTBR < CALLVR, the FERRF flag in CASTR is set to 1 because the clock frequency is erroneous. If the FERRIE bit in CAICR is 1, a frequency error interrupt is generated. Also, the MENDF flag in CASTR is set to 1. If the MENDIE bit in CAICR is 1, a measurement end interrupt is generated.
(6) While the CFME bit in CACR0 is 1, the counter value is retained in CACNTBR and compared with the values of CAULVR and CALLVR every time a valid edge is input. Writing 0 to the CFME bit in CACR0 clears the counter and stops up-counting.

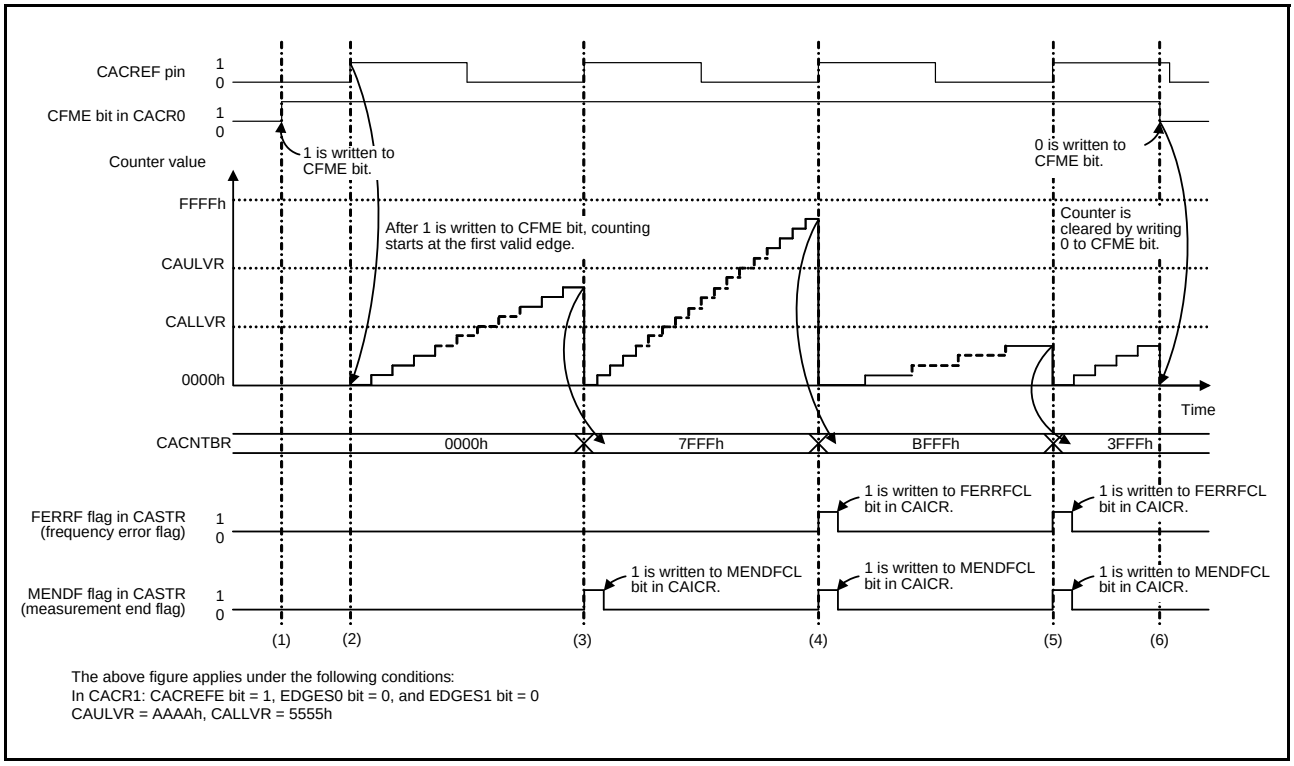



**Figure 10.2 Operating Example of Clock Frequency Accuracy Measurement Circuit Based on CACREF Pin Input**

### 10.3.2 Measuring Clock Frequency Based on Another Clock Source

Figure 10.3 shows an operating example of the clock frequency accuracy measurement circuit based on another clock source.

The clock frequency accuracy measurement circuit operates as shown below when measuring the clock frequency.

(1) When 1 is written to the CFME bit in CACR0 with the RPS bit in CACR2 set to 1, clock frequency measurement based on another clock source is enabled.

(2) After 1 is written to the CFME bit, the timer starts up-counting when the valid edge selected by the EDGES[1:0] bits in CACR1 is input based on the clock source selected by the RSCS[2:0] bits in CACR2.

(3) When the next valid edge is input, the counter value is retained in CACNTBR and compared with the values of CAULVR and CALLVR. If both CACNTBR ≤ CAULVR and CACNTBR ≥ CALLVR are satisfied, only the MENDF flag in CASTR is set to 1 because the clock frequency is correct. If the MENDIE bit in CAICR is 1, a measurement end interrupt is generated.

(4) When the next valid edge is input, the counter value is retained in CACNTBR and compared with the values of CAULVR and CALLVR. In the case of CACNTBR > CAULVR, the FERRF flag in CASTR is set to 1 because the clock frequency is erroneous. If the FERRIE bit in CAICR is 1, a frequency error interrupt is generated. Also, the MENDF flag in CASTR is set to 1. If the MENDIE bit in CAICR is 1, a measurement end interrupt is generated.

(5) When the next valid edge is input, the counter value is retained in CACNTBR and compared with the values of CAULVR and CALLVR. In the case of CACNTBR < CALLVR, the FERRF flag in CASTR is set to 1 because the clock frequency is erroneous. If the FERRIE bit in CAICR is 1, a frequency error interrupt is generated. Also, the MENDF flag in CASTR is set to 1. If the MENDIE bit in CAICR is 1, a measurement end interrupt is generated.

(6) While the CFME bit in CACR0 is 1, the counter value is retained in CACNTBR and compared with the values of CAULVR and CALLVR every time a valid edge is input. Writing 0 to the CFME bit in CACR0 clears the counter and stops up-counting.

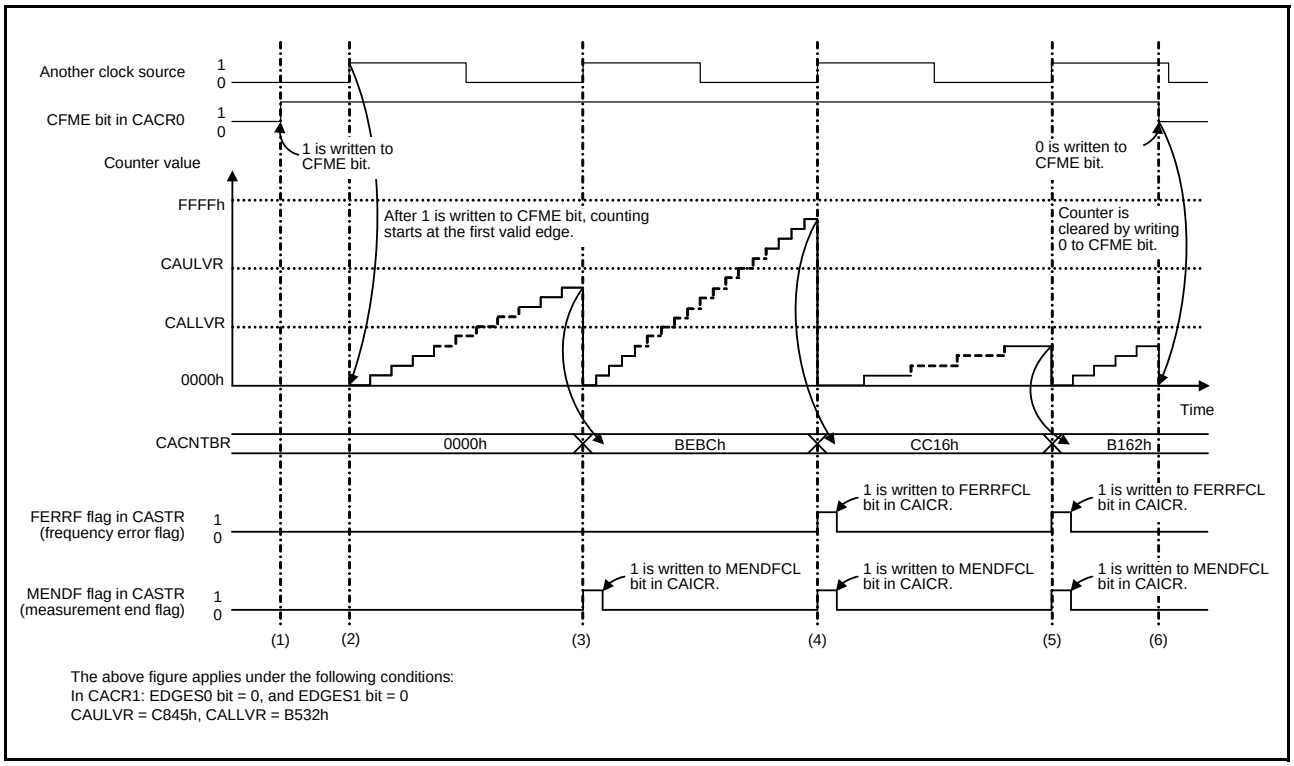



**Figure 10.3 Operating Example of Clock Frequency Accuracy Measurement Circuit Based on Another Clock Source**

### 10.3.3 Digital Filtering of Signals on the CACREF Pin

The CACREF pin has a digital filter. Levels on the target pin for sampling are conveyed to the internal circuitry after matching three times at the selected sampling interval and the same level continues to be conveyed internally until the level on the pin again matches three times.
Enabling and disabling of the digital filter and its sampling clock are selectable.
The counter value held in CACNTBR may be in error by up to one cycle of the sampling clock due to the difference between the phases of the digital filter and the signal input to the CACREF pin.
When a frequency dividing clock is selected as a count source clock, the counter value error is obtained by the following formula:
Counter value error = (One cycle of the count source clock) / (One cycle of the sampling clock)

## 10.4 Interrupt Requests

The clock frequency accuracy measurement circuit issues three types of interrupt request: frequency error interrupt, measurement end interrupt, and overflow interrupt. When an interrupt source is generated, the corresponding status flag becomes 1. Table 10.3 lists details on the interrupt requests of the clock frequency accuracy measurement circuit.

**Table 10.3 Interrupt Requests of Clock Frequency Accuracy Measurement Circuit**

| Interrupt Request | Interrupt Enable Bit | Status Flag | Interrupt Source |
|---|---|---|---|
| Frequency error interrupt | CAICR.FERRIE | CASTR.FERRF | The result of comparing CACNTBR to CAULVR and CALLVR is either CACNTBR > CAULVR or CACNTBR < CALLVR. |
| Measurement end interrupt | CAICR.MENDIE | CASTR.MENDF | A valid edge is input from the CACREF pin.<br>Note however that a measurement end interrupt does not occur at the first valid edge after writing 1 to the CFME bit in CACR0. |
| Overflow interrupt | CAICR.OVFIE | CASTR.OVFF | The counter has overflowed. |

## 10.5 Usage Notes

### 10.5.1 Module Stop Function Setting

CAC operation can be disabled or enabled using module stop control register C (MSTPCRC). The initial setting is for the CAC to be halted. Register access is enabled by canceling the module stop state. For details, refer to section 11, Low Power Consumption.

# 11. Low Power Consumption

## 11.1 Overview

This MCU has several functions for reducing power consumption, by setting clock dividers, stopping modules, changing to low power consumption mode in normal operation, and changing to operating power control mode.
Table 11.1 lists the specifications of low power consumption functions, and Table 11.2 lists the conditions to change to low power consumption modes, states of the CPU and peripheral modules, and the method for exiting each mode.
After a reset, this MCU returns to normal mode, but modules except the DTC and RAM are stopped.

**Table 11.1 Specifications of Low Power Consumption Functions**

| Item | Specification |
|---|---|
| Clock divider functions | The frequency division ratio can be set independently for the system clock (ICLK), peripheral module clock (PCLKB), S12AD clock (PCLKD), and FlashIF clock (FCLK). *1 |
| Module stop function | Each peripheral module can be stopped independently by Module Stop Control Register |
| Low power consumption modes | Power consumption can be reduced by selecting an appropriate power consumption mode according to which module is need to operate. *2<br>Three low power consumption modes are available *3<br>• Sleep mode<br>• Deep sleep mode<br>• Software standby mode |
| Operating power control modes | Power consumption can be reduced in normal operation, sleep mode, and deep sleep mode by selecting an appropriate operating power control mode according to the operating frequency and operating voltage.<br>Three operating power control modes are available<br>• High-speed operating mode<br>• Middle-speed operating mode<br>• Low-speed operating mode |

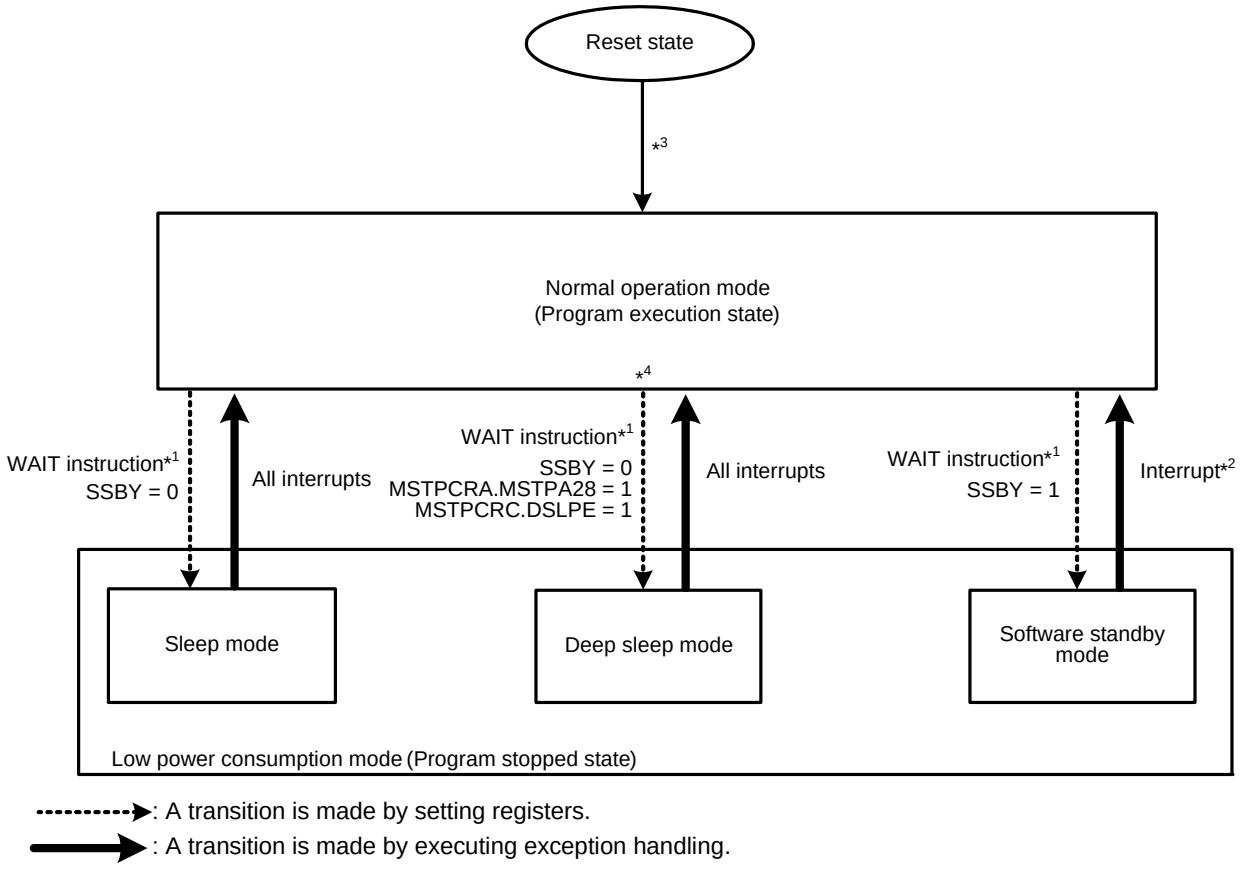
Note 1. For details, refer to section 9, Clock Generation Circuit.
Note 2. For details, refer to section 11.6, Low Power Consumption Modes.
Note 3. For details, refer to Figure 11.2, Operating Conditions of Each Power Consumption Mode.

**Table 11.2 Operating Conditions of Each Power Consumption Mode**

| | Sleep Mode | Deep Sleep Mode | Software Standby Mode |
|---|---|---|---|
| Entry trigger | Control register + instruction | Control register + instruction | Control register + instruction |
| Exit trigger | Interrupt | Interrupt | Interrupt*1 |
| After exiting from each mode, CPU begins from*2 | Interrupt handling | Interrupt handling | Interrupt handling |
| Main clock oscillator | Operating possible | Operating possible | Stopped |
| Sub-clock oscillator | Operating possible | Operating possible | Operating possible |
| High-speed on-chip oscillator | Operating possible | Operating possible | Stopped |
| Low-speed on-chip oscillator | Operating possible | Operating possible | Stopped |
| IWDT-dedicated on-chip oscillator | Operating possible*3 | Operating possible*3 | Operating possible*3 |
| PLL | Operating possible | Operating possible | Stopped |
| CPU | Stopped (Retained) | Stopped (Retained) | Stopped (Retained) |
| RAM0 (0000 0000h to 0000 3FFFh) | Operating possible (Retained) | Stopped (Retained) | Stopped (Retained) |
| DTC | Operating possible*5 | Stopped (Retained) | Stopped (Retained) |
| Flash memory | Operating | Stopped (Retained) | Stopped (Retained) |
| Independent watchdog timer (IWDT) | Operating possible*3 | Operating possible*3 | Operating possible*3 |
| Realtime clock (RTC) | Operating possible | Operating possible | Operating possible |
| Voltage detection circuit (LVD) | Operating possible | Operating possible | Operating possible |
| Power-on reset circuit | Operating | Operating | Operating |
| Peripheral modules | Operating possible | Operating possible | Stopped (Retained)*4 |
| I/O ports | Operating | Operating | Retained |
| RTCOUT | Operating possible | Operating possible | Operating possible |
| CLKOUT | Operating possible | Operating possible | Operating possible*6 |

“Operating possible” means that operating or stopped can be controlled by the register setting.
“Stopped (Retained)” means that internal register values are retained and internal operations are suspended.

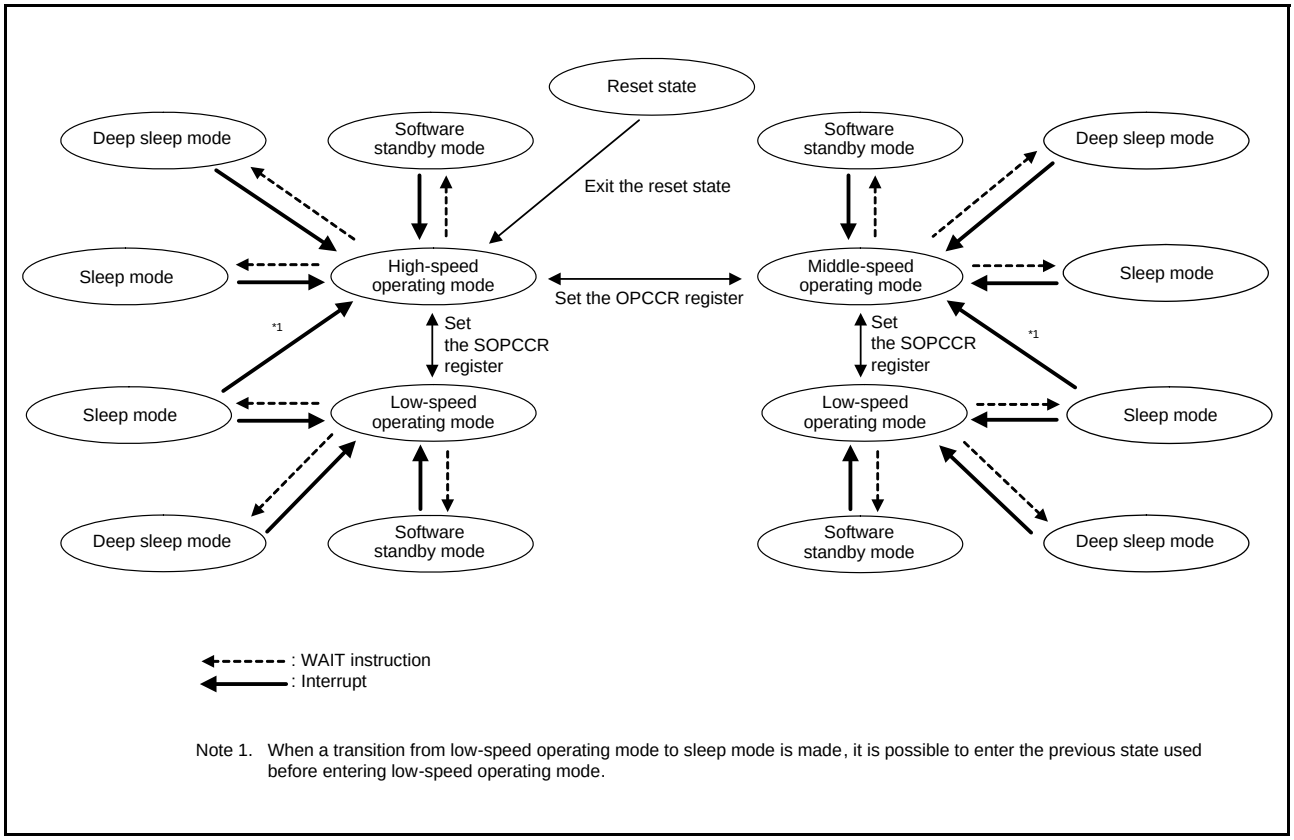
Note 1. “Interrupts” here indicates an external pin interrupt (the NMI or IRQ0 to IRQ7) or any of peripheral interrupts (the RTC alarm, RTC interval, IWDT, voltage monitoring, and USB interrupts).
Note 2. This does not include a RES# pin reset, power-on reset, voltage monitoring reset, or independent watchdog-timer reset. One of these reset sources initiate transition to reset state.
Note 3. Operating or stopping is selected by setting the IWDT sleep mode count stop control bit (IWDTSLCSTP) in option function select register 0 (OFS0) in IWDT auto-start mode. In any mode other than IWDT auto-start mode, operating or stopping is selected by the setting of the sleep mode count stop control bit (SLCSTP) in the IWDT count stop control register (IWDTCSTPR).
Note 4. The peripheral logic states are retained.
Note 5. During sleep mode, do not write to the system control related registers (indicated by ‘SYSTEM’ in the Module Symbol column in Table 5.1, List of I/O Registers (Address Order).
Note 6. Stopped when the clock output source select bits (CKOCR.CKOSEL[2:0]) are set to a value other than 011b (sub-clock oscillator).




Note 1. If an interrupt is accepted while entering low power consumption mode, the transition to the program stop state is exited and the interrupt exception handling is executed.
Note 2. “Interrupts” here indicates an external pin interrupt (the NMI or IRQ0 to IRQ7) or any of peripheral interrupts (the RTC alarm, RTC interval, IWDT, voltage monitoring, and USB interrupts).
Note 3. The LOCO is the clock source following a transition from the reset state to normal mode.
Note 4. Makes a transition from sleep mode, deep sleep mode, or software standby mode to normal operating mode by an interrupt. In the case of exiting sleep mode, the clock source after exiting is selectable. For details, refer to the description of the RSTCKCR register.
For software standby mode, the clock source after exiting is the same as that of the clock before entering software standby mode.

Transition to the reset state is made at any state instantly after a RES# pin reset, power on reset, voltage monitoring reset, IWDT reset, or software reset is generated.

**Figure 11.1 Mode Transitions**




Note 1. When a transition from low-speed operating mode to sleep mode is made, it is possible to enter the previous state used before entering low-speed operating mode.

**Figure 11.2 Operating Modes**

- The sub-clock oscillator does not stop when entering software standby mode.
- It is possible to return from sleep mode to the previous operating state used before entering sleep mode. However, when a transition from low-speed operating mode to sleep mode is made, it is possible to enter the previous state used before entering low-speed operating mode.
- After exiting the reset state, operation starts in high-speed operating mode.

**Table 11.3 Oscillator Usability in Each Mode**

| | PLL | HOCO | LOCO | Main Clock Oscillator | Sub-Clock Oscillator |
|---|---|---|---|---|---|
| High-speed operating mode | Usable*1 | Usable | Usable | Usable | Usable |
| Middle-speed operating mode | Usable*1 | Usable | Usable | Usable | Usable |
| Low-speed operating mode | Not usable | Not usable | Not usable | Not usable | Usable |

Note 1. The PLL is usable when the power supply voltage is 2.4 V or above.

## 11.2 Register Descriptions

### 11.2.1 Standby Control Register (SBYCR)

Address(es): 0008 000Ch

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | SSBY | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b14 to b0 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |
| b15 | SSBY | Software Standby | 0: Set entry to sleep mode or deep sleep mode after the WAIT instruction is executed<br>1: Set entry to software standby mode after the WAIT instruction is executed | R/W |

Note: • Set the PRCR.PRC1 bit to 1 (write enabled) before rewriting this register.

**SSBY Bit (Software Standby)**

The SSBY bit specifies the transition destination after the WAIT instruction is executed.
When the SSBY bit is set to 1, the MCU enters software standby mode after execution of the WAIT instruction. When the MCU returns to normal mode after an interrupt has triggered and exits from software standby mode, the SSBY bit remains 1. The SSBY bit can be cleared by writing 0 to the SSBY bit.

When the oscillation stop detection function enable bit (OSTDCR.OSTDE) in the oscillation stop detection control register is 1, the set value of the SSBY bit is invalid. Even if the SSBY bit is 1, the MCU will enter sleep mode or deep sleep mode after execution of the WAIT instruction.

## 11.2.2 Module Stop Control Register A (MSTPCRA)

Address(es): 0008 0010h

| | b31 | b30 | b29 | b28 | b27 | b26 | b25 | b24 | b23 | b22 | b21 | b20 | b19 | b18 | b17 | b16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | MSTPA28 | — | — | — | — | — | — | — | — | MSTPA19 | — | MSTPA17 | — |
| Value after reset: | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | MSTPA15 | — | — | — | — | — | MSTPA9 | — | — | — | — | — | — | — | — | — |
| Value after reset: | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b8 to b0 | — | Reserved | These bits are read as 1. The write value should be 1. | R/W |
| b9 | MSTPA9 | Multifunction Timer Pulse Unit 2 Module Stop | Target module: MTU (MTU0 to MTU5)<br>0: This module clock is enabled<br>1: This module clock is disabled | R/W |
| b14 to b10 | — | Reserved | These bits are read as 1. The write value should be 1. | R/W |
| b15 | MSTPA15 | Compare Match Timer (Unit 0) Module Stop | Target module: CMT unit 0 (CMT0, CMT1)<br>0: This module clock is enabled<br>1: This module clock is disabled | R/W |
| b16 | — | Reserved | This bit is read as 1. The write value should be 1. | R/W |
| b17 | MSTPA17 | 12-Bit A/D Converter Module Stop | Target module: S12AD<br>0: This module clock is enabled<br>1: This module clock is disabled | R/W |
| b18 | — | Reserved | This bit is read as 1. The write value should be 1. | R/W |
| b19 | MSTPA19 | D/A Converter Module Stop | Target module: DA<br>0: This module clock is enabled<br>1: This module clock is disabled | R/W |
| b27 to b20 | — | Reserved | These bits are read as 1. The write value should be 1. | R/W |
| b28 | MSTPA28 | Data Transfer Controller Module Stop | Target module: DTC<br>0: This module clock is enabled<br>1: This module clock is disabled | R/W |
| b31 to b29 | — | Reserved | These bits are read as 1. The write value should be 1. | R/W |

Note: • Set the PRCR.PRC1 bit to 1 (write enabled) before rewriting this register.

## 11.2.3 Module Stop Control Register B (MSTPCRB)

Address(es): 0008 0014h

| | b31 | b30 | b29 | b28 | b27 | b26 | b25 | b24 | b23 | b22 | b21 | b20 | b19 | b18 | b17 | b16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | MSTPB30 | — | — | — | MSTPB26 | — | — | MSTPB23 | — | MSTPB21 | — | MSTPB19 | — | MSTPB17 | — |
| Value after reset: | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | MSTPB9 | — | — | MSTPB6 | — | MSTPB4 | — | — | — | — |
| Value after reset: | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b3 to b0 | — | Reserved | These bits are read as 1. The write value should be 1. | R/W |
| b4 | MSTPB4 | Serial Communication Interface SCIf Module Stop | Target module: SCIf (SCI12)<br>0: This module clock is enabled<br>1: This module clock is disabled | R/W |
| b5 | — | Reserved | This bit is read as 1. The write value should be 1. | R/W |
| b6 | MSTPB6 | DOC Module Stop | Target module: DOC<br>0: This module clock is enabled<br>1: This module clock is disabled | R/W |
| b8, b7 | — | Reserved | These bits are read as 1. The write value should be 1. | R/W |
| b9 | MSTPB9 | ELC Module Stop | Target module: ELC<br>0: This module clock is enabled<br>1: This module clock is disabled | R/W |
| b16 to b10 | — | Reserved | These bits are read as 1. The write value should be 1. | R/W |
| b17 | MSTPB17 | Serial Peripheral Interface 0 Module Stop | Target module: RSPI0<br>0: This module clock is enabled<br>1: This module clock is disabled | R/W |
| b18 | — | Reserved | This bit is read as 1. The write value should be 1. | R/W |
| b19 | MSTPB19 *1 | USB0 Module Stop | Target module: USB0<br>0: This module clock is enabled<br>1: This module clock is disabled | R/W |
| b20 | — | Reserved | This bit is read as 1. The write value should be 1. | R/W |
| b21 | MSTPB21 | $I^2C$ Bus Interface 0 Module Stop | Target module: RIIC0<br>0: This module clock is enabled<br>1: This module clock is disabled | R/W |
| b22 | — | Reserved | This bit is read as 1. The write value should be 1. | R/W |
| b23 | MSTPB23 | CRC Calculator Module Stop | Target module: CRC<br>0: This module clock is enabled<br>1: This module clock is disabled | R/W |
| b25, b24 | — | Reserved | These bits are read as 1. The write value should be 1. | R/W |
| b26 | MSTPB26 | Serial Communication Interface 5 Module Stop | Target module: SCI5<br>0: This module clock is enabled<br>1: This module clock is disabled | R/W |
| b29 to b27 | — | Reserved | These bits are read as 1. The write value should be 1. | R/W |
| b30 | MSTPB30 | Serial Communication Interface 1 Module Stop | Target module: SCI1<br>0: This module clock is enabled<br>1: This module clock is disabled | R/W |
| b31 | — | Reserved | This bit is read as 1. The write value should be 1. | R/W |

Note: • Set the PRCR.PRC1 bit to 1 (write enabled) before rewriting this register.
Note 1. For entering software standby mode after rewriting this bit, wait for two UCLK cycles after rewriting, and execute the WAIT instruction.

## 11.2.4 Module Stop Control Register C (MSTPCRC)

Address(es): 0008 0018h

| | b31 | b30 | b29 | b28 | b27 | b26 | b25 | b24 | b23 | b22 | b21 | b20 | b19 | b18 | b17 | b16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | DSLPE | — | — | — | — | — | — | — | — | — | — | — | MSTPC19 | — | — | — |
| Value after reset: | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | MSTPC0 |
| Value after reset: | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b0 | MSTPC0 | RAM0 Module Stop*1 | Target module: RAM0 (0000 0000h to 0000 FFFFh)<br>0: RAM0 operating<br>1: RAM0 stopped | R/W |
| b18 to b1 | — | Reserved | These bits are read as 1. The write value should be 1. | R/W |
| b19 | MSTPC19 | Clock Frequency Accuracy Measurement Circuit Module Stop*2 | Target module: CAC<br>0: This module clock is enabled<br>1: This module clock is disabled | R/W |
| b30 to b20 | — | Reserved | These bits are read as 1. The write value should be 1. | R/W |
| b31 | DSLPE | Deep Sleep Mode Enable | 0: Deep sleep mode is disabled<br>1: Deep sleep mode is enabled | R/W |

Note: • Set the PRCR.PRC1 bit to 1 (write enabled) before rewriting this register.

Note 1. The MSTPC0 bit should not be set to 1 during access to the corresponding RAM. The corresponding RAM should not be accessed while the MSTPC0 bit is set to 1.

Note 2. The MSTPC19 bit should be rewritten while the oscillation of the clock to be controlled by this bit is stable. For entering software standby mode after rewriting this bit, wait for two cycles of the slowest clock among the clocks output by the oscillators actually oscillating and execute the WAIT instruction.

### DSLPE Bit (Deep Sleep Mode Enable)

The DSLPE bit enables or disables a transition to deep sleep mode.

When the CPU executes the WAIT instruction with the DSLPE bit set to 1 and the SBYCR.SSBY and MSTPCRA.MSTPA28 bits meet specified conditions, the MCU enters deep sleep mode. For details, refer to section 11.6.2, Deep Sleep Mode.

## 11.2.5 Operating Power Control Register (OPCCR)

Address(es): 0008 00A0h

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | — | OPCMTSF | — | OPCM[2:0] | | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b2 to b0 | OPCM[2:0] | Operating Power Control Mode Select | b2 b0<br>0 0 0: High-speed operating mode<br>0 1 0: Middle-speed operating mode<br>Settings other than above are prohibited. | R/W |
| b3 | — | Reserved | This bit is read as 0. The write value should be 0. | R/W |
| b4 | OPCMTSF | Operating Power Control Mode Transition Status Flag | 0: Transition completed<br>1: During transition | R |
| b7 to b5 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |

Note: • Set the PRCR.PRC1 bit to 1 (write enabled) before rewriting this register.

The OPCCR register is used to reduce power consumption in normal operating mode, sleep mode, and deep sleep mode. Power consumption can be reduced according to the operating frequency and operating voltage to be used by the OPCCR setting.

The OPCCR register cannot be rewritten under the following conditions:

- When the OPCCR.OPCMTSF flag is 1 (during transition)
- Time period from WAIT instruction execution for a sleep mode transition, until exit from sleep mode to normal operation
- Time period from WAIT instruction execution for a deep sleep mode transition, until exit from deep sleep mode to normal operation
- When the SOPCCR.SOPCM bit is 1 (low-speed operating mode)

The OPCCR register cannot be rewritten while the flash memory is being programmed or erased (P/E).
For the procedures of changing operating power control modes, refer to Function in section 11.5, Function for Lower Operating Power Consumption.
During sleep mode or mode transitions, do not write to the registers related to system control (indicated by 'SYSTEM' in the Module Symbol column in Table 5.1, List of I/O Registers (Address Order)).

### OPCM[2:0] Bits (Operating Power Control Mode Select)

The OPCM[2:0] bits select operating power control mode in normal operating mode, sleep mode, and deep sleep mode. Table 11.4 shows the relationship between operating power control modes, the OPCM[2:0] and SOPCM bit settings, and the operating frequency and voltage ranges.

**OPCMTSF Flag (Operating Power Control Mode Transition Status Flag)**

This flag indicates the switching control state during and after operating power mode transition.

This flag becomes 1 when the value of the OPCM[2:0] bits is rewritten, and 0 when mode transition is completed. Read this flag and confirm that it is 0 before proceeding to the next processing. Only rewrite the OPCM[2:0] bits when this flag is 0.

## 11.2.6 Sub Operating Power Control Register (SOPCCR)

Address(es): 0008 00AAh

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | — | SOPCMTSF | — | — | — | SOPCM |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b0 | SOPCM | Sub Operating Power Control Mode Select | 0: High-speed operating mode or middle-speed operating mode*1<br>1: Low-speed operating mode | R/W |
| b3 to b1 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |
| b4 | SOPCMTSF | Sub Operating Power Control Mode Transition Status Flag | 0: Transition completed<br>1: During transition | R |
| b7 to b5 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |

Note: • Set the PRCR.PRC1 bit to 1 (write enabled) before rewriting this register.
Note 1. Depends on the setting of OPCCR.OPCM[2:0].

The SOPCCR register is used to reduce power consumption in normal operating mode, sleep mode, and deep sleep mode by controlling a transition to low-speed operating mode.
Setting this register initiates entry to/exit from low-speed operating mode.
Low-speed operating mode is used for the sub-clock oscillator only.
The OPCCR register cannot be rewritten when the SOPCM bit is 1 (low-speed operating mode).

The SOPCCR register cannot be rewritten under the following conditions:

- When the SOPCCR.SOPCMTSF bit is 1 (during transition)
- Time period from WAIT instruction execution for a sleep mode transition, until exit from sleep mode to normal operation
- Time period from WAIT instruction execution for a deep sleep mode transition, until exit from deep sleep mode to normal operation

This register cannot be rewritten while the flash memory is being programmed or erased (P/E).
For the procedures for changing operating power control modes, refer to Function in section 11.5, Function for Lower Operating Power Consumption.
During sleep mode or mode transitions, do not write to the registers related to system control (indicated by 'SYSTEM' in the Module Symbol column in Table 5.1, List of I/O Registers (Address Order)).

### SOPCM Bit (Sub Operating Power Control Mode Select)

The SOPCM bit selects operating power control in normal operating mode and sleep mode.
Setting this bit to 1 allows a transition to low-speed operating mode. Setting this bit to 0 allows a return to the operating mode (operating mode set by OPCCR.OPCM[2:0]) before the transition to low-speed operating mode.
Table 11.4 shows the relationship between operating power control modes, the OPCM[2:0] and SOPCM bit settings, and the operating frequency and voltage ranges.

### SOPCMTSF Flag (Sub Operating Power Control Mode Transition Status Flag)

The SOPCMTSF flag indicates the switching control state when the sub operating power control mode is switched. This flag becomes 1 when the value of the SOPCM bit is rewritten, and 0 when mode transition is completed. Read this flag and confirm that it is 0 before proceeding to the next processing. Only rewrite the SOPCM bit when this flag is 0.

**Table 11.4 Operating Frequency and Voltage Ranges in Operating Power Control Modes**

| Operating Power Control Mode | OPCM [2:0] Bits | SOPCM Bit | Operating Voltage Range | Operating Frequency Range | | | | |
|---|---|---|---|---|---|---|---|---|
| | | | | Flash Memory Read Frequency | | | | Flash Memory Programming/ Erasure Frequency |
| | | | | ICLK | FCLK | PCLKD | PCLKB | FCLK |
| High-speed operating mode | 000b | 0 | 2.7 to 3.6 V | Up to 32 MHz | Up to 32 MHz | Up to 32 MHz | Up to 32 MHz | 1 MHz to 32 MHz |
| | | | 2.4 to 2.7 V | Up to 16 MHz | Up to 16 MHz | Up to 16 MHz | Up to 16 MHz | — |
| | | | 1.8 to 2.4 V | Up to 8 MHz | Up to 8 MHz | Up to 8 MHz | Up to 8 MHz | — |
| Middle-speed operating mode | 010b | 0 | 2.4 to 3.6 V | Up to 12 MHz | Up to 12 MHz | Up to 12 MHz | Up to 12 MHz | 1 MHz to 12 MHz |
| | | | 1.8 to 2.4 V | Up to 8 MHz | Up to 8 MHz | Up to 8 MHz | Up to 8 MHz | 1 MHz to 8 MHz |
| Low-speed operating mode | 000b | 1 | 1.8 to 3.6 V | Up to 32.768 kHz | Up to 32.768 kHz | Up to 32.768 kHz | Up to 32.768 kHz | — |
| | 010b | 1 | 1.8 to 3.6 V | | | | | |

Note: • When using the FCLK at lower than 4 MHz during programming or erasing the flash memory, the frequency can be set to 1, 2, or 3 MHz.

Each operating power control mode is described below.

- High-Speed Operating Mode

The maximum operating frequency during FLASH read is 32 MHz for ICLK, FCLK, PCLKD, and PCLKB. The operating voltage range is 1.8 to 3.6 V during FLASH read. However, for ICLK, FCLK, PCLKD, and PCLKB, the maximum operating frequency during FLASH read is 16 MHz when the operating voltage is 2.4 V or larger and smaller than 2.7 V. The maximum operating frequency during FLASH read is 8 MHz for all the clocks when the operating voltage is 1.8 V or larger and smaller than 2.4 V.
During FLASH programming/erasure, the operating frequency range is 1 to 32 MHz and the operating voltage range is 2.7 to 3.6 V.
After a reset is canceled, operation is started from this mode.
The following restriction applies when middle-speed operating mode is selected.

- The PLL can be used when the operating voltage is 2.4 V or above.

Figure 11.3 shows the operating voltages and frequencies in high-speed operating mode.

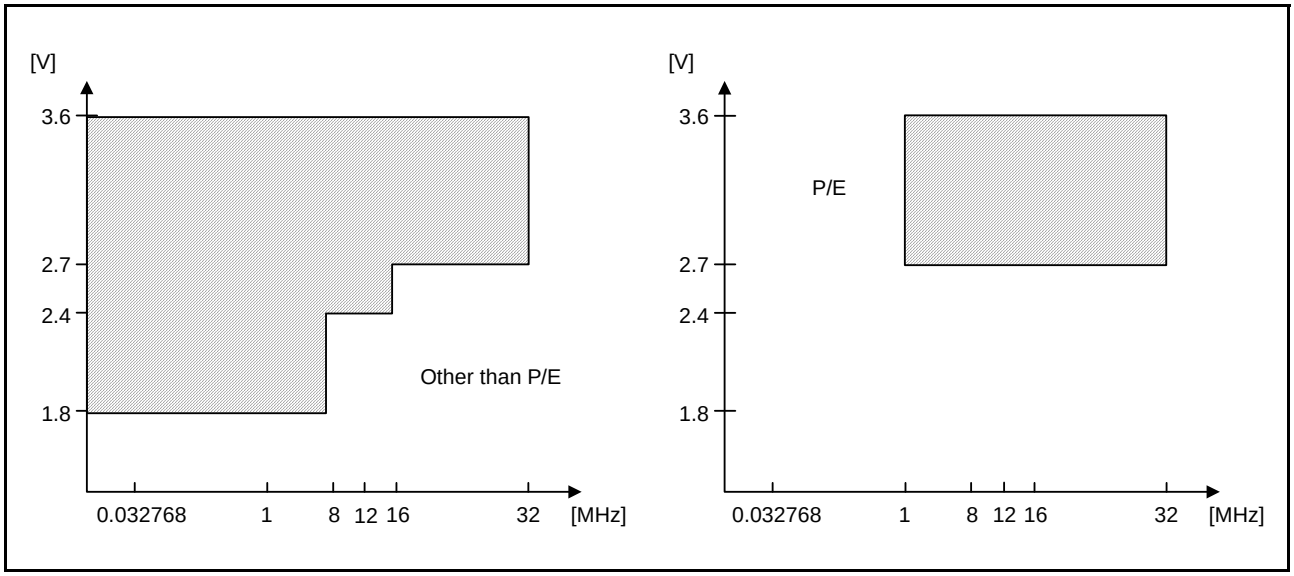



**Figure 11.3 Operating Voltages and Frequencies in High-Speed Operating Mode**

Note: • When using the FCLK at lower than 4 MHz during programming or erasing the flash memory, the frequency can be set to 1, 2, or 3 MHz.

- Middle-Speed Operating Mode

As compared to high-speed operating mode, this mode reduces power consumption for low-speed operation.
The maximum operating frequency during FLASH read is 12 MHz for ICLK, FCLK, PCLKD, and PCLKB. The operating voltage range is 1.8 to 3.6 V during FLASH read. The maximum operating frequency during FLASH read is 8 MHz for all the clocks when the operating voltage is 1.8 V or larger and smaller than 2.4 V.
During FLASH programming/erasure, the operating frequency range is 1 to 12 MHz and the operating voltage range is 1.8 to 3.6 V. The maximum operating frequency during FLASH programming/erasure is 8 MHz when the operating voltage is 1.8 V or larger and smaller than 2.4 V.
The power consumption of this mode is lower than that of high speed mode under the same conditions.
The following restriction applies when middle-speed operating mode is selected:

- The PLL can be used when the operating voltage is 2.4 V or above.

Figure 11.4 shows the operating voltages and frequencies in middle-speed operating mode.

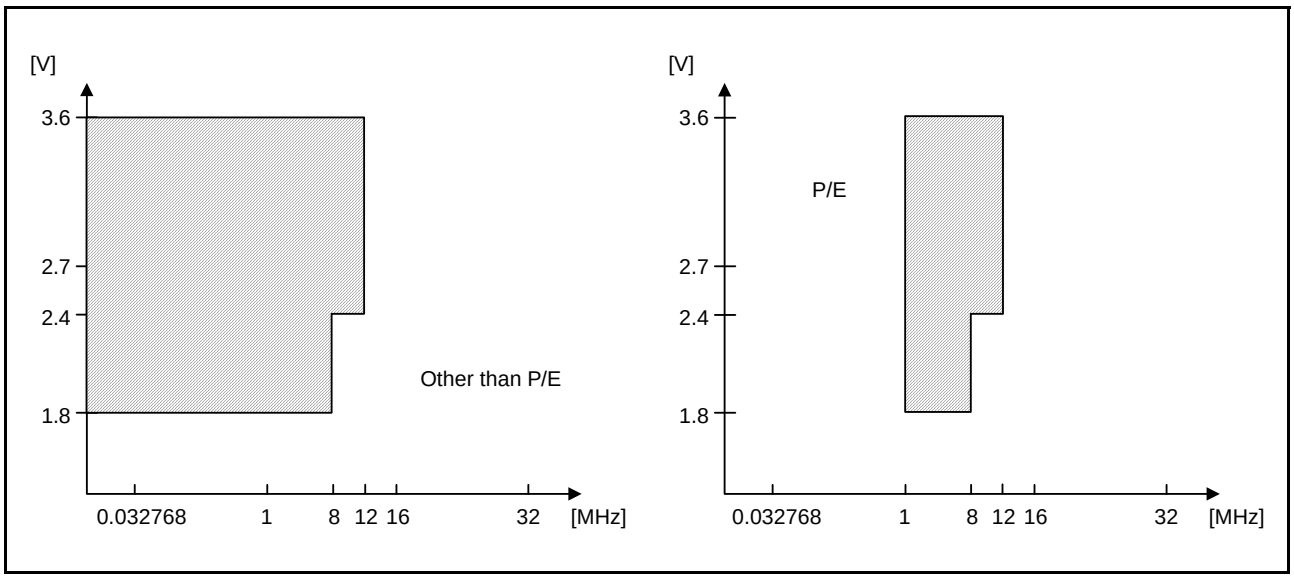



**Figure 11.4 Operating Voltages and Frequencies in Middle-Speed Operating Mode**

Note: • When using the FCLK at lower than 4 MHz during programming or erasing the flash memory, the frequency can be set to 1, 2, or 3 MHz.

- Low-Speed Operating Mode

A transition to low-speed operating mode is set by writing 1 to the SOPCM bit in the SOPCCR register. The setting of the OPCM[2:0] bits cannot be modified during low-speed operating mode. This mode is used only for the sub oscillator of 32.768 kHz.

During reading the flash memory (FLASH), the maximum operating frequency of ICLK, FCLK, PCLKD, and PCLKB is 32.768 kHz. The operating voltage is in the range of 1.8 to 3.6 V.

The following restrictions apply when low-speed operating mode is selected:

- P/E operations for flash memory are prohibited.
- The PLL, main clock oscillator, LOCO, and HOCO cannot be used.

Note: • The SOPCM bit cannot be set to 1 when the PLLCR2.PLLEN bit is 0 (PLL is operating).
The SOPCM bit cannot be set to 1 when the HOCOCR.HCSTP bit is 0 (HOCO is operating).
The SOPCM bit cannot be set to 1 when the MOSCCR.MOSTP bit is 0 (MOSC is operating).
The SOPCM bit cannot be set to 1 when the LOCOCR.LCSTP bit is 0 (LOCO is operating).

Figure 11.5 shows the operating voltages and frequencies in low-speed operating mode.

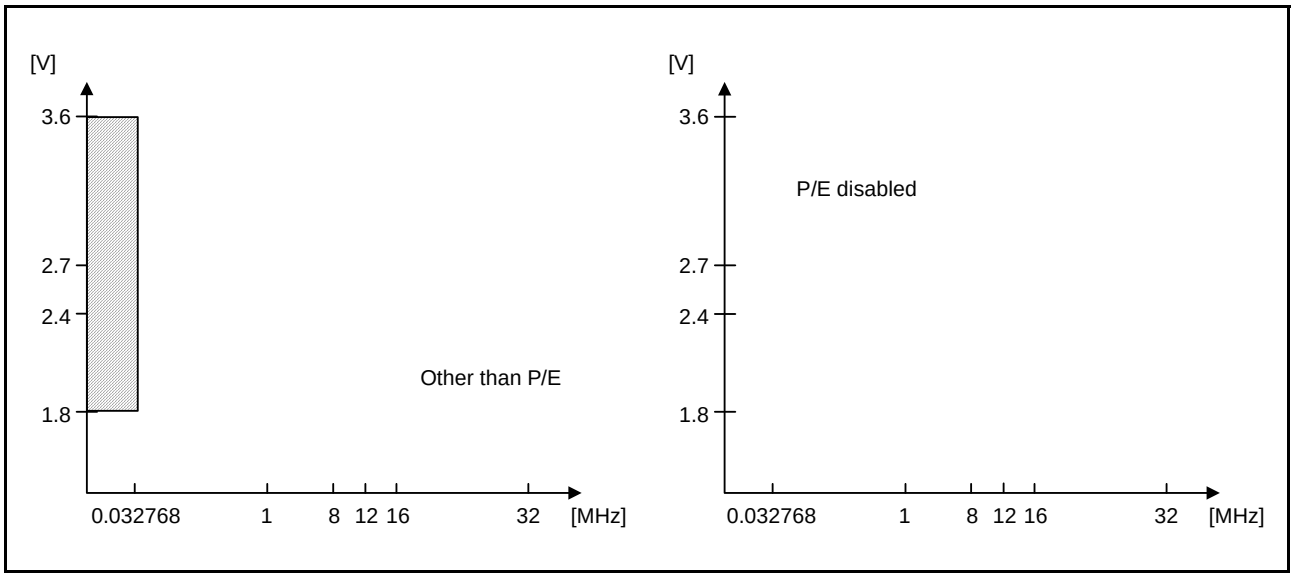



**Figure 11.5 Operating Voltages and Frequencies in Low-Speed Operating Mode**

## 11.2.7 Sleep Mode Return Clock Source Switching Register (RSTCKCR)

Address(es): 0008 00A1h

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | RSTCKEN | — | — | — | — | RSTCKSEL[2:0] | | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b2 to b0 | RSTCKSEL [2:0] | Sleep Mode Return Clock Source Select | b2 b0<br>0 0 0: LOCO is selected<br>0 0 1: HOCO is selected*1<br>0 1 0: Main clock oscillator is selected<br>Settings other than above are prohibited when the RSTCKEN bit is 1. | R/W |
| b6 to b3 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |
| b7 | RSTCKEN | Sleep Mode Return Clock Source Switching Enable | 0: Clock source switching at exit from sleep mode is disabled<br>1: Clock source switching at exit from sleep mode is enabled | R/W |

Note: • Set the PRCR.PRC1 bit to 1 (write enabled) before rewriting this register.
Note 1. HOCO can only be selected when entering high-speed operating mode.

RSTCKCR is used to control clock source switching at exit from sleep mode.
When exit from sleep mode is initiated by setting RSTCKCR, the main clock oscillator stop bit in the main clock oscillator control register (MOSCCR.MOSTP), the HOCO stop bit in the high-speed on-chip oscillator control register (HOCOCR.HCSTP), and the LOCO stop bit in the low-speed on-chip oscillator control register (LOCOCR.LCSTP) are automatically modified to the operating state corresponding to the clock source to be used after transition. The value of the RSTCKSEL[2:0] bits is automatically reloaded to the clock source select bits in system clock control register 3 (SCKCR3.CKSEL[2:0]).

### RSTCKSEL[2:0] Bits (Sleep Mode Return Clock Source Select)

The RSTCKSEL[2:0] bits select the clock source to be used at exit from sleep mode.
The clock source selected by the RSTCKSEL[2:0] bits is enabled only when the RSTCKEN bit is 1.
As shown in Figure 11.2, Operating Modes, when returning from sleep mode to high-speed operating mode, the LOCO, HOCO, or main clock oscillator can be selected. When returning from sleep mode to middle-speed operating mode, the LOCO or main clock oscillator can be selected. However, in this case, the frequency of each clock (ICLK, FCLK, PCLKD, and PCLKB) must be lower than 12 MHz when the power supply voltage is 2.4 V or above, and lower than 8 MHz when the voltage is below 2.4 V.

**Table 11.5 When Exiting Sleep Mode to High-Speed Operating Mode and Middle-Speed Operating Mode**

| Operating Mode during Sleep | Clock Source during Sleep | RSTCKSEL | Operating Mode after Exiting | Clock Source after Exiting |
|---|---|---|---|---|
| High-speed operating mode or low-speed operating mode after exit from high-speed operating mode | Sub-clock oscillator | 000b (LOCO) | High-speed operating mode | LOCO |
| | | 001b (HOCO) | | HOCO |
| | | 010b (main clock oscillator) | | Main clock oscillator |
| Middle-speed operating mode or low-speed operating mode after exit from middle-speed operating mode | Sub-clock oscillator | 000b (LOCO) | Middle-speed operating mode | LOCO |
| | | 010b (main clock oscillator) | | Main clock oscillator*1 |

Note 1. The frequency of each clock (ICLK, FCLK, PCLKD, and PCLKB) must be lower than 12 MHz when the power supply voltage is 2.4 V or above, and lower than 8 MHz when the voltage is below 2.4 V.

**RSTCKEN Bit (Sleep Mode Return Clock Source Switching Enable)**

The RSTCKEN bit enables or disables clock source switching when sleep mode is exited.

When returning from sleep mode while this bit is enabled, the SOPCM bit in the SOPCCR register is automatically rewritten to 0 (middle-speed operating mode or high-speed operating mode).

The value of the frequency division setting (in the SCKCR register) is retained.

To exit sleep mode to middle-speed operating mode when the main clock oscillator is selected, the frequency of each clock must be lower than 12 MHz when the power supply voltage is 2.4 V or above, and lower than 8 MHz when the voltage is below 2.4 V.

## 11.3 Reducing Power Consumption by Switching Clock Signals

The clock frequency can change by setting the SCKCR.FCK[3:0], ICK[3:0], PCKB[3:0], and PCKD[3:0] bits. The CPU, DTC, ROM, and RAM clocks can be set by the ICK[3:0] bits. The peripheral module clocks can be set by the PCKB[3:0] and PCKD[3:0] bits.
The flash memory clock can be set by the FCK[3:0] bits.
For details, refer to section 9, Clock Generation Circuit.

## 11.4 Module Stop Function

The module stop function can be set for each on-chip peripheral module.
When the MSTPmi bit (m = A to C, i = 0 to 31) in MSTPCRA to MSTPCRC is set to 1, the specified module stops operating and enters the module stop state, but the CPU continues to operate independently. When the corresponding MSTPmi bit is cleared to 0, the module exits the module state and restarts operating at the end of the bus cycle. The internal states of modules are retained in the module stop state.
After a reset is canceled, all modules other than the DTC, and on-chip RAM are in the module stop state. Basically the registers in the module stop state cannot be read or written. However, note that data may be written to these registers if write access is made immediately after the setting of the module stop state. To avoid this, always write to the module stop registers after confirming that the last register setting is done.

## 11.5 Function for Lower Operating Power Consumption

By selecting an appropriate operating power control mode according to the operating frequency and operating voltage, power consumption can be reduced in normal mode, sleep mode, and deep sleep mode.

### 11.5.1 Setting Operating Power Control Mode

Examples of the procedures for switching operating power control modes are shown below:

(1) Switching from Normal Power Consumption Mode to Low Power Consumption Mode

- Example 1: From high-speed operating mode to middle-speed operating mode

(High-speed operation in high-speed operating mode)
↓
Set the frequency of each clock to lower than the maximum operating frequency for middle-speed operating mode
↓
Confirm that the OPCCR.OPCMTSF flag is 0 (transition completed)
↓
Set the OPCCR.OPCM[2:0] bits to 010b (middle-speed operating mode)
↓
Confirm that the OPCCR.OPCMTSF flag is 0 (transition completed)
↓
(Middle-speed operation in middle-speed operating mode)

- Example 2: From high-speed/middle-speed operating mode to low-speed operating mode

(High-speed operation in high-speed operating mode/middle-speed operation in middle-speed operating mode)
↓
Set the frequency of each clock to lower than the maximum operating frequency for low-speed operating mode
↓
Confirm that all clock sources but the sub-clock oscillator are stopped
↓
Confirm that the SOPCCR.SOPCMTSF flag is 0 (transition completed)
↓
Set the SOPCCR.SOPCM bit to 1 (low-speed operating mode)
↓
Confirm that the SOPCCR.SOPCMTSF flag is 0 (transition completed)
↓
Low-speed operation in low-speed operating mode

(2) Switching from Low Power Consumption Mode to Normal Power Consumption Mode

- Example 1: From low-speed operating mode to high-speed/middle-speed operating mode

Low-speed operation in low-speed operating mode
↓
Confirm that the SOPCCR.SOPCMTSF flag is 0 (transition completed)
↓
Set the SOPCCR.SOPCM bit to 0 (high-speed operating mode or middle-speed operating mode)
↓
Confirm that the SOPCCR.SOPCMTSF flag is 0 (transition completed)
↓
Set the frequency of each clock to lower than the maximum operating frequency for high-speed/middle-speed operating mode
↓
(High-speed operation in high-speed operating mode/middle-speed operation in middle-speed operating mode)

- Example 2: From middle-speed operating mode to high-speed operating mode

Middle-speed operation in middle-speed operating mode
↓
Confirm that the OPCCR.OPCMTSF flag is 0 (transition completed)
↓
Set the OPCCR.OPCM[2:0] bit to 0 (high-speed operating mode)
↓
Confirm that the OPCCR.OPCMTSF flag is 0 (transition completed)
↓
Set the frequency of each clock to lower than the maximum operating frequency for high-speed operating mode
↓
High-speed operation in high-speed operating mode

## 11.6 Low Power Consumption Modes

### 11.6.1 Sleep Mode

#### 11.6.1.1 Entry to Sleep Mode

When the WAIT instruction is executed while the SBYCR.SSBY bit is 0, the CPU enters sleep mode. In sleep mode, the CPU stops operating but the contents of its internal registers are retained. Other peripheral functions do not stop.
Counting by the IWDT stops if a transition to sleep mode is made while the IWDT is being used in auto-start mode and the OFS0.IWDTSLCSTP bit is 1. In the same way, counting by the IWDT stops if a transition to sleep mode is made while the IWDT is being used in register start mode and the IWDTCSTPR.SLCSTP bit is 1.
Furthermore, counting by the IWDT continues if a transition to sleep mode is made while the IWDT is being used in auto-start mode and the OFS0.IWDTSLCSTP bit is 0 (counting by the IWDT continues through transitions to low power consumption modes). In the same way, counting by the IWDT continues if a transition to sleep mode is made while the IWDT is being used in register start mode and the IWDTCSTPR.SLCSTP bit is 0.
To use sleep mode, make the following settings and then execute a WAIT instruction.

(1) Clear the PSW.I bit*1 of the CPU to 0.
(2) Set the interrupt destination to be used for exit from sleep mode.
(3) Set the priority*2 of the interrupt to be used for exit from sleep mode to a level higher than the setting of the PSW.IPL[3:0] bits*1 of the CPU.
(4) Set the IERm.IENj bit*2 to 1 for the interrupt.
(5) Read the I/O register that is written last and confirm that the written value has been reflected.
(6) Execute the WAIT instruction (this automatically sets the I bit*1 in the PSW of the CPU to 1).

Note 1. For details, refer to section 2, CPU.
Note 2. For details, refer to section 14, Interrupt Controller (ICUb).

### 11.6.1.2 Exit from Sleep Mode

Exit from sleep mode is initiated by any interrupt, a RES# pin reset, a power-on reset, a voltage monitoring reset, or a reset caused by an IWDT underflow.

- Initiated by an interrupt
  An interrupt initiates exit from sleep mode and the interrupt exception handling starts. If a maskable interrupt has been masked by the CPU (the priority level*1 of the interrupt has been set to a value lower than that of the PSW.IPL[3:0] bits*2 of the CPU), sleep mode is not exited.
- Initiated by a RES# pin reset
  When the RES# pin is driven low, the MCU enters the reset state. When the RES# pin is driven high after the reset signal is input for a predetermined time period, the CPU starts the reset exception handling.
- Initiated by a power-on reset
  A power-on reset asserts a reset to the MCU.
  When a power-on reset is negated by a rise in the supply voltage, the CPU starts the reset exception handling.
- Initiated by a voltage monitoring reset
  A voltage monitoring reset asserts a reset to the MCU.
  When a voltage monitoring reset is negated by a rise in the supply voltage, the CPU starts the reset exception handling.
- Initiated by an independent watchdog timer reset
  An internal reset generated by an IWDT underflow asserts a reset to the MCU. However, when IWDT counting is stopped in sleep mode by setting OFS0.IWDTSTRT = 0 and OFS0.IWDTSLCSTP = 1, or OFS0.IWDTSTRT = 1 and IWDTCSTPR.SLCSTP = 1, the IWDT is stopped in sleep mode and sleep mode is not exited by the independent watchdog timer reset.

Note 1. For details, refer to section 14, Interrupt Controller (ICUb).
Note 2. For details, refer to section 2, CPU.

### 11.6.1.3 Sleep Mode Return Clock Source Switching Function

To switch the clock source used for exit from sleep mode, set the sleep mode return clock source switching register (RSTCKCR) and the wait control register for each clock. When the return interrupt is generated, after oscillation settling of the oscillator specified as the return clock, the clock source is automatically switched, and then operation exits sleep mode. At this time, the registers related to clock source switching are automatically rewritten.
For details, refer to section 11.2.7, Sleep Mode Return Clock Source Switching Register (RSTCKCR).
For details on settings the oscillation stabilization wait time, refer to section 9.2.13, Main Clock Oscillator Wait Control Register (MOSCWTCR) and section 9.2.14, High-Speed On-Chip Oscillator Wait Control Register (HOCOWTCR).

## 11.6.2 Deep Sleep Mode

### 11.6.2.1 Entry to Deep Sleep Mode

When a WAIT instruction is executed with the MSTPCRC.DSLPE bit set to 1, the MSTPCRA.MSTPA28 bit set to 1, and the SBYCR.SSBY bit cleared to 0, a transition to deep sleep mode is made.
In deep sleep mode, the CPU and the DTC, ROM, and RAM clocks stop. Peripheral functions do not stop.
Counting by the IWDT stops if a transition to deep sleep mode is made while the IWDT is being used in auto-start mode and the OFS0.IWDTSLCSTP bit is 1. In the same way, counting by the IWDT stops if a transition to deep sleep mode is made while the IWDT is being used in register start mode and the IWDTCSTPR.SLCSTP bit is 1.
Furthermore, counting by the IWDT continues if a transition to deep sleep mode is made while the IWDT is being used in auto-start mode and the OFS0.IWDTSLCSTP bit is 0 (counting by the IWDT continues through transitions to low power consumption modes). In the same way, counting by the IWDT continues if a transition to deep sleep mode is made while the IWDT is being used in register start mode and the IWDTCSTPR.SLCSTP bit is 0.

To use deep sleep mode, make the following settings and then execute a WAIT instruction.

(1) Clear the PSW.I bit*1 of the CPU to 0.
(2) Set the interrupt destination to be used for exit from deep sleep mode.
(3) Set the priority*2 of the interrupt to be used for exit from deep sleep mode to a level higher than the setting of the PSW.IPL[3:0] bits*1 of the CPU.
(4) Set the IERm.IENj bit*2 to 1 for the interrupt.
(5) Read the I/O register that is written last and confirm that the written value has been reflected.
(6) Execute a WAIT instruction (executing a WAIT instruction causes automatic setting of the PSW.I bit*1 of the CPU to 1).

Note 1. For details, refer to section 2, CPU.
Note 2. For details, refer to section 14, Interrupt Controller (ICUb).

#### 11.6.2.2 Exit from Deep Sleep Mode

Exit from deep sleep mode is initiated by any interrupt, a RES# pin reset, a power-on reset, a voltage monitoring reset, or a reset caused by an IWDT underflow.

- Initiated by an interrupt
  An interrupt initiates exit from deep sleep mode and the interrupt exception handling starts. If a maskable interrupt has been masked by the CPU (the priority level*1 of the interrupt has been set to a value lower than that of the PSW.IPL[3:0] bits*2 of the CPU), deep sleep mode is not exited.
- Initiated by the RES# pin reset
  When the RES# pin is driven low, the MCU enters the reset state. When the RES# pin is driven high after the reset signal is input for a predetermined time period, the CPU starts the reset exception handling.
- Initiated by a power-on reset
  A power-on reset asserts a reset to the MCU.
  When a power-on reset is negated by a rise in the supply voltage, the CPU starts the reset exception handling.
- Initiated by a voltage monitoring reset
  A voltage monitoring reset asserts a reset to the MCU.
  When a voltage monitoring reset is negated by a rise in the supply voltage, the CPU starts the reset exception handling.
- Initiated by the independent watchdog timer
  An internal reset generated by an IWDT underflow asserts a reset to the MCU. However, when IWDT counting is stopped in deep sleep mode by setting OFS0.IWDTSTRT = 0 and OFS0.IWDTSLCSTP = 1, or OFS0.IWDTSTRT = 1 and IWDTCSTPR.SLCSTP = 1, the IWDT is stopped in deep sleep mode and deep sleep mode is not exited by the independent watchdog timer reset.

Note 1. For details, refer to section 14, Interrupt Controller (ICUb).
Note 2. For details, refer to section 2, CPU.

## 11.6.3 Software Standby Mode

### 11.6.3.1 Entry to Software Standby Mode

When a WAIT instruction is executed with the SBYCR.SSBY bit set to 1, a transition to software standby mode is made. In this mode, the CPU, on-chip peripheral functions, and all the other functions except the sub-clock oscillator stop. However, the contents of the CPU internal registers, RAM data, the states of on-chip peripheral functions, the I/O ports, and the sub-clock oscillator are retained. Software standby mode allows significant reduction in power consumption because the oscillator stops in this mode.
Clear the DTCST.DTCST bit to 0 before executing the WAIT instruction.
Counting by the IWDT stops if a transition to software standby mode is made while the IWDT is being used in auto-start mode and the OFS0.IWDTSLCSTP bit is 1. In the same way, counting by the IWDT stops if a transition to software standby mode is made while the IWDT is being used in register start mode and the IWDTCSTPR.SLCSTP bit is 1.

Furthermore, counting by the IWDT continues if a transition to software standby mode is made while the IWDT is being used in auto-start mode and the OFS0.IWDTSLCSTP bit is 0 (counting by the IWDT continues through transitions to low power consumption modes). In the same way, counting by the IWDT continues if a transition to software standby mode is made while the IWDT is being used in register start mode and the IWDTCSTPR.SLCSTP bit is 0.
To use software standby mode, make the following settings and then execute a WAIT instruction.

(1) Clear the PSW.I bit*1 of the CPU to 0.
(2) Set the interrupt destination to be used for recovery from software standby mode to the CPU.
(3) Set the priority*2 of the interrupt to be used for recovery from software standby mode to a level higher than the setting of the PSW.IPL[3:0] bits*1 of the CPU.
(4) Set the IERm.IENj bit*2 to 1 for the interrupt.
(5) Read the I/O register that is written last and confirm that the written value has been reflected.
(6) Execute a WAIT instruction (executing a WAIT instruction causes automatic setting of the PSW.I bit*1 of the CPU to 1).

Note 1. For details, refer to section 2, CPU.
Note 2. For details, refer to section 14, Interrupt Controller (ICUb).

#### 11.6.3.2 Exit from Software Standby Mode

Exit from software standby mode is initiated by an external pin interrupt (the NMI or IRQ0 to IRQ7), peripheral interrupts (the RTC alarm, RTC interval, IWDT, voltage monitoring, and USB interrupts), a RES# pin reset, a power-on reset, a voltage monitoring reset, or an independent watchdog timer reset. When any trigger which initiates exit from software standby mode is asserted, the oscillators which were operating before entry to software standby mode restart operation. After the oscillation of all these oscillators has been stabilized, operation returns from software standby mode.

- Initiated by an interrupt
  When an interrupt request from among the NMI, IRQ0 to IRQ7, RTC alarm, RTC interval, IWDT, voltage monitoring, and USB interrupts is generated, each of the oscillators which was operating before the transition to software standby mode resumes oscillation. After the oscillation stabilization wait time of each oscillator set by the MOSCWTCR.MSTS[4:0] bits or HOCOWTCR.HSTS[4:0] bits has elapsed, the MCU exits software standby mode and interrupt exception processing starts.
- Initiated by a RES# pin reset
  Clock oscillation starts when the low level is applied to the RES# pin. Clock supply for the MCU starts at the same time. Keep the level on the RES# pin low over the time required for oscillation of the clocks to become stable. Reset exception processing starts when the high level is applied to the RES# pin.
- Initiated by a power-on reset
  A power-on reset asserts a reset to the MCU.
  When a power-on reset is negated by a rise in the supply voltage, the CPU starts the reset exception handling.
- Initiated by a voltage monitoring reset
  A voltage monitoring reset asserts a reset to the MCU.
  When a voltage monitoring reset is negated by a rise in the supply voltage, the CPU starts the reset exception handling.
- Initiated by an independent watchdog timer reset
  An internal reset generated by an IWDT underflow asserts a reset to the MCU.
  Note that the independent watchdog timer is stopped in software standby mode due to the register settings (OFS0.IWDTSTRT = 0 and OFS0.IWDTSLCSTP = 1, or OFS0.IWDTSTRT = 1 and IWDTCSTPR.SLCSTP = 1) in software standby mode. In that case, exit from software standby mode by the independent watchdog timer reset cannot be done.

### 11.6.3.3 Example of Software Standby Mode Application

Figure 11.6 shows an example of entry to software standby mode by the falling edge of the IRQn pin, and exit from software standby mode by the rising edge of the IRQn pin.

In this example, an IRQn interrupt is accepted with the IRQCRi.IRQMD[1:0] bits of the ICU set to 01b (falling edge), and then the IRQCRi.IRQMD[1:0] bits are set to 10b (rising edge). After that, the SBYCR.SSBY bit is set to 1 and the WAIT instruction is executed. Thus entry to software standby mode is completed. After that, exit from software standby mode is initiated by the rising edge of the IRQn pin.

To exit software standby mode, settings of the interrupt controller (ICU) are also necessary. For details, refer to section 14, Interrupt Controller (ICUb).

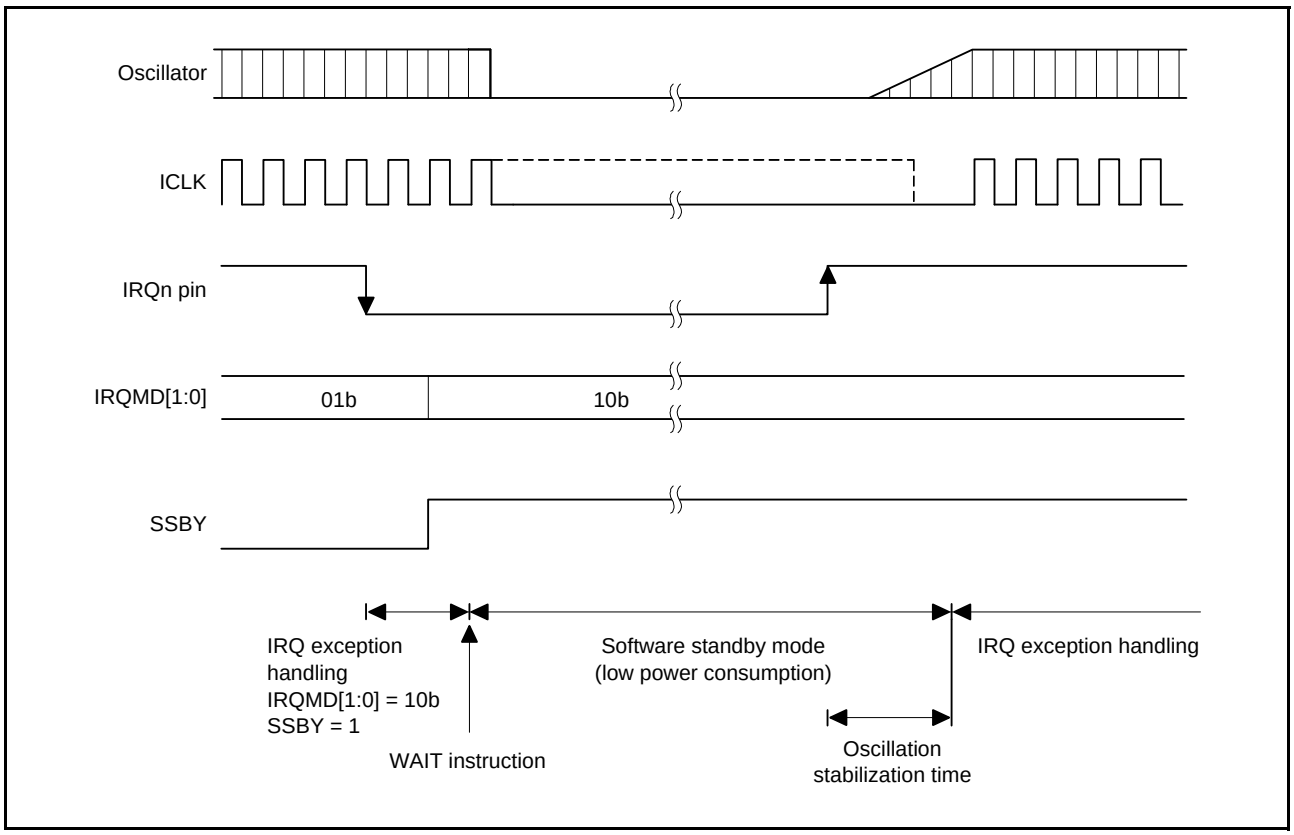



**Figure 11.6 Example of Software Standby Mode Application**

## 11.7 Usage Notes

### 11.7.1 I/O Port States

I/O port states are retained in software standby mode. Therefore, the supply current is not reduced if output signals are high level.

### 11.7.2 Module Stop State of DTC

Before setting the MSTPCRA.MSTPA28 bit to 1, clear the DTCST.DTCST bit of the DTC to 0 to avoid activating the DTC module.
For details, refer to section 16, Data Transfer Controller (DTCa).

### 11.7.3 On-Chip Peripheral Module Interrupts

Interrupts do not operate in the module stop state. Therefore, if the module stop state is made after an interrupt request is generated, a CPU interrupt source or a DTC startup source cannot be cleared. For this reason, disable interrupts before entering the module stop state.

### 11.7.4 Write Access to MSTPCRA, MSTPCRB, and MSTPCRC

Write accesses to MSTPCRA, MSTPCRB, and MSTPCRC should be made only by the CPU.

### 11.7.5 Timing of WAIT Instructions

The WAIT instruction is executed before completion of the preceding register write. The WAIT instruction being executed before the register setting is modified may cause unintended operation. To avoid this, always execute the WAIT instruction after confirming that the last register setting is done.

### 11.7.6 Rewrite the Register by DTC in Sleep Mode

Depending on the settings of the OFS0.IWDTSLCSTP bit and IWDTCSTPR.SLCSTP bit, the IWDT may also stop in sleep mode. To avoid this, do not set up the DTC to rewrite any registers related to the IWDT in sleep mode.
The RSTCKCR register is a register that switches the clock source at exit from sleep mode. Changing the RSTCKCR register in sleep mode causes unintended operation, so do not write to this register in sleep mode.

# 12. Register Write Protection Function

The register write protection function protects important registers from being overwritten for in case a program runs out of control. The registers to be protected are set with the protect register (PRCR).

Table 12.1 lists the association between the PRCR bits and the registers to be protected.

**Table 12.1 Association between PRCR Bits and Registers to be Protected**

| PRCR Bit | Register to be Protected |
|---|---|
| PRC0 | • Registers related to the clock generation circuit:<br>SCKCR, SCKCR3, PLLCR, PLLCR2, MOSCCR, SOSCCR, LOCOCR, ILOCOCR, HOCOCR, OSTDCR, OSTDSR, CKOCR |
| PRC1 | • Register related to the operating modes:<br>SYSCR1<br>• Registers related to low power consumption functions:<br>SBYCR, MSTPCRA, MSTPCRB, MSTPCRC, OPCCR, RSTCKCR, SOPCCR<br>• Registers related to the clock generation circuit:<br>MOFCR, MOSCWTCR<br>• Software reset register:<br>SWRR |
| PRC2 | • Registers related to the clock generation circuit:<br>HOCOWTCR |
| PRC3 | • Registers related to the LVD:<br>LVCMPCR, LVDLVLR, LVD1CR0, LVD1CR1, LVD1SR, LVD2CR0, LVD2CR1, LVD2SR |

## 12.1 Register Descriptions

### 12.1.1 Protect Register (PRCR)

Address(es): 0008 03FEh

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | PRKEY[7:0] | | | | | | | | — | — | — | — | PRC3 | PRC2 | PRC1 | PRC0 |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Function | R/W |
|---|---|---|---|---|
| b0 | PRC0 | Protect Bit 0 | Enables writing to the registers related to the clock generation circuit.<br>0: Write disabled<br>1: Write enabled | R/W |
| b1 | PRC1 | Protect Bit 1 | Enables writing to the registers related to operating modes, low power consumption functions, the clock generation circuit, and software reset.<br>0: Write disabled<br>1: Write enabled | R/W |
| b2 | PRC2 | Protect Bit 2 | Enables writing to the registers related to the clock generation circuit.<br>0: Write disabled<br>1: Write enabled | R/W |
| b3 | PRC3 | Protect Bit 3 | Enables writing to the registers related to the LVD.<br>0: Write disabled<br>1: Write enabled | R/W |
| b7 to b4 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |
| b15 to b8 | PRKEY[7:0] | PRC Key Code | These bits control permission and prohibition of writing to the PRCR register.<br>To modify the PRCR register, write A5h to the 8 higher-order bits and the desired value to the 8 lower-order bits as a 16-bit unit. | R/W*1 |

Note 1. Write data is not retained.

**PRCi Bits (Protect Bit i) (i = 0 to 3)**

These bits enable or disable writing to the corresponding registers to be protected.

Setting the PRCi bits to 1 and 0 enable and disable writing to the corresponding registers to be protected, respectively.

# 13. Exception Handling

## 13.1 Exception Events

During execution of a program by the CPU, the occurrence of a certain event may cause execution of that program to be suspended and execution of another program to be started. Such kinds of events are called exception events.
The RX CPU supports six types of exceptions. The types of exception events are shown in Figure 13.1.
The occurrence of an exception causes the processor mode to switch to supervisor mode.

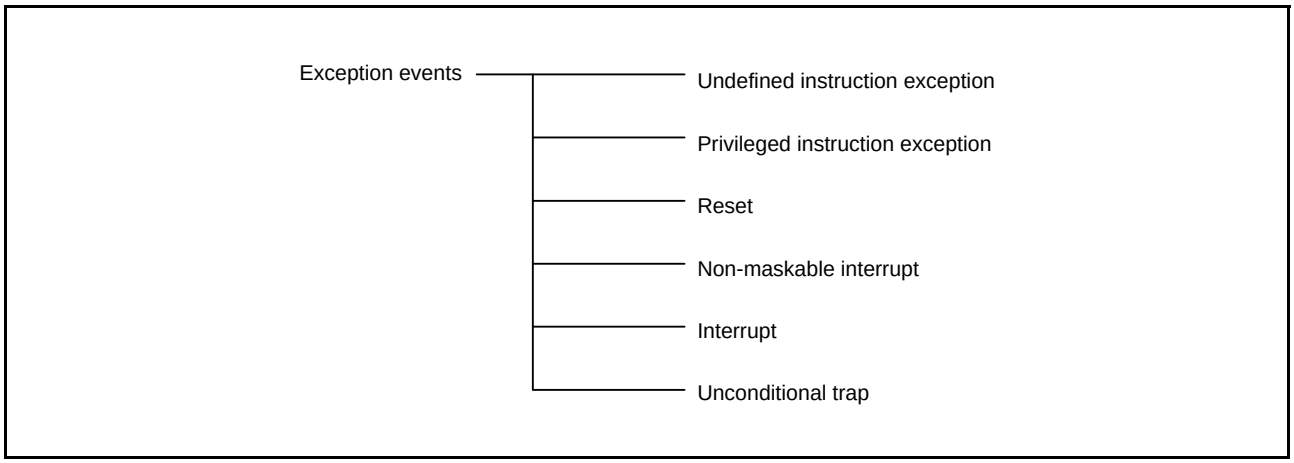



**Figure 13.1 Types of Exception Events**

### 13.1.1 Undefined Instruction Exception

An undefined instruction exception occurs when execution of an undefined instruction (an instruction not implemented) is detected.

### 13.1.2 Privileged Instruction Exception

A privileged instruction exception occurs when execution of a privileged instruction is detected in user mode. Privileged instructions can be executed only in supervisor mode.

### 13.1.3 Reset

A reset is generated by input of a reset signal to the CPU. This has the highest priority of any exception and is always accepted.

### 13.1.4 Non-Maskable Interrupt

A non-maskable interrupt is generated by input of a non-maskable interrupt signal to the CPU and is only used when a fatal fault is considered to have occurred in the system. Never use the non-maskable interrupt with an attempt to return to the program that was being executed at the time of interrupt generation after the exception handling routine is ended.

### 13.1.5 Interrupts

Interrupts are generated by the input of interrupt signals to the CPU. A fast interrupt can be selected as the interrupt with the highest priority. In the case of the fast interrupt, hardware pre-processing and hardware post-processing are handled fast. The priority level of the fast interrupt is 15 (the highest). The exception handling of interrupts is masked when the I bit in PSW is 0.

### 13.1.6 Unconditional Trap

An unconditional trap is generated when the INT or BRK instruction is executed.

## 13.2 Exception Handling Procedure

In the exception handling, part of the processing is handled automatically by hardware and part of it is handled by a program (exception handling routine) that has been written by the user. Figure 13.2 shows the processing procedure when an exception other than a reset is accepted.

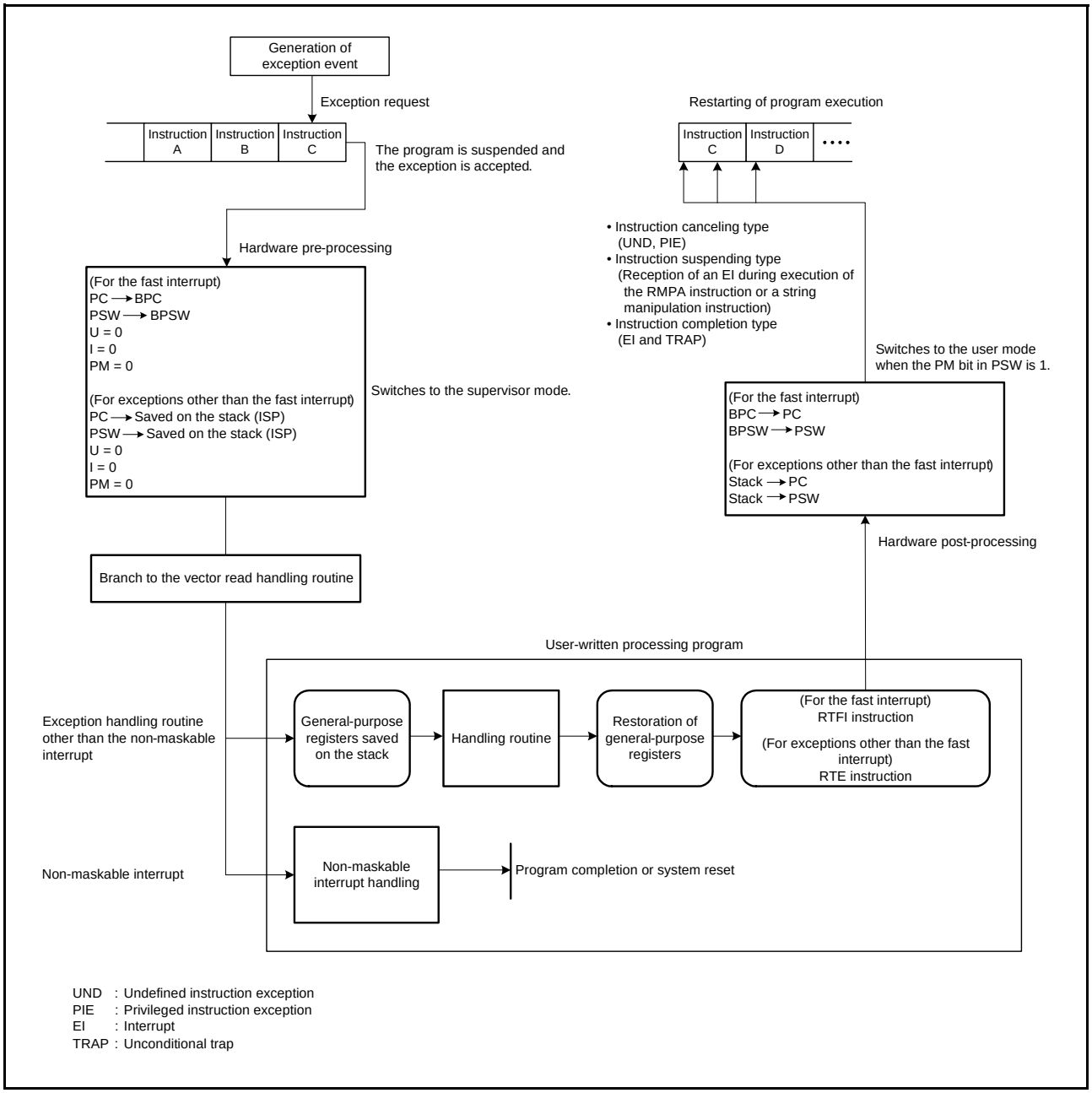



**Figure 13.2 Outline of Exception Handling Procedure**

When an exception is accepted, hardware processing by the RX CPU is followed by access to the vector to acquire the address of the branch destination. In the vector, a vector address is allocated to each exception, and the branch destination address of the exception handling routine is written to each vector address.
Hardware pre-processing by the RX CPU handles saving of the contents of the program counter (PC) and processor status word (PSW). In the case of a fast interrupt, the contents are saved in the backup PC (BPC) and the backup PSW (BPSW), respectively. In the case of exceptions other than a fast interrupt, the contents are saved in the stack area.
General purpose registers and control registers other than the PC and PSW that are to be used within the exception handling routine must be saved on the stack by a user program at the start of the exception handling routine.
On completion of processing by an exception handling routine, execution is restored from the exception handling routine to the original program by saving the registers saved on the stack and executing the RTE instruction. For return from a fast interrupt, the RTFI instruction is used instead. In the case of a non-maskable interrupt, however, finish the program or reset the system without returning to the original program.
Hardware post-processing by the RX CPU handles restoration of the contents of PC and PSW. In the case of a fast interrupt, the values of BPC and BPSW are restored to PC and PSW, respectively. In the case of exceptions other than a fast interrupt, the values are restored from the stack to PC and PSW.

## 13.3 Acceptance of Exception Events

When an exception occurs, the CPU suspends the execution of the program and processing branches to the exception handling routine.

### 13.3.1 Acceptance Timing and Saved PC Value

Table 13.1 lists the timing of acceptance and the program counter (PC) value to be saved for each exception event.

**Table 13.1 Acceptance Timing and Saved PC Value**

| Exception Event | | Type of Handling | Acceptance Timing | Value Saved in BPC or on the Stack |
|---|---|---|---|---|
| Undefined instruction exception | | Instruction canceling type | During instruction execution | PC value of the instruction that generated the exception |
| Privileged instruction exception | | Instruction canceling type | During instruction execution | PC value of the instruction that generated the exception |
| Reset | | Instruction abandonment type | Any machine cycle | None |
| Non-maskable interrupt | During execution of the RMPA, SCMPU, SMOVB, SMOVF, SMOVU, SSTR, SUNTIL, and SWHILE instructions | Instruction suspending type | During instruction execution | PC value of the instruction being executed |
| | Other than above | Instruction completion type | At the next break between instructions | PC value of the next instruction |
| Interrupt | During execution of the RMPA, SCMPU, SMOVB, SMOVF, SMOVU, SSTR, SUNTIL, and SWHILE instructions | Instruction suspending type | During instruction execution | PC value of the instruction being executed |
| | Other than above | Instruction completion type | At the next break between instructions | PC value of the next instruction |
| Unconditional trap | | Instruction completion type | At the next break between instructions | PC value of the next instruction |

### 13.3.2 Vector and Site for Saving the Values in the PC and PSW

The vector for each type of exception and the site for saving the values of the program counter (PC) and processor status word (PSW) are listed in Table 13.2.

**Table 13.2 Vector and Site for Saving the Values in the PC and PSW**

| Exception | | Vector | Site for Saving the Values in the PC and PSW |
|---|---|---|---|
| Undefined instruction exception | | Fixed vector table | Stack |
| Privileged instruction exception | | Fixed vector table | Stack |
| Reset | | Fixed vector table | Nowhere |
| Non-maskable interrupt | | Fixed vector table | Stack |
| Interrupt | Fast interrupt | FINTV | BPC and BPSW |
| | Other than above | Relocatable vector table (INTB) | Stack |
| Unconditional trap | | Relocatable vector table (INTB) | Stack |

## 13.4 Hardware Processing for Accepting and Returning from Exceptions

This section describes the hardware processing for accepting and returning from exceptions other than a reset.

### (1) Hardware Pre-Processing for Accepting an Exception

#### (a) Saving PSW

- For a fast interrupt
  PSW → BPSW
- For exceptions other than a fast interrupt
  PSW → Stack

#### (b) Updating PM, U, and I Bits in PSW

I: Set to 0
U: Set to 0
PM: Set to 0

#### (c) Saving PC

- For a fast interrupt
  PC → BPC
- For exceptions other than a fast interrupt
  PC → Stack

#### (d) Setting Branch Destination Address of Exception Handling Routine in PC

Processing is shifted to the exception handling routine by acquiring the vector corresponding to the exception and then branching accordingly.

### (2) Hardware Post-Processing for Execution of RTE and RTFI Instructions

#### (a) Restoring PSW

- For a fast interrupt
  BPSW → PSW
- For exceptions other than a fast interrupt
  Stack → PSW

#### (b) Restoring PC

- For a fast interrupt
  BPC → PC
- For exceptions other than a fast interrupt
  Stack → PC

## 13.5 Hardware Pre-Processing

The hardware pre-processing from reception of each exception request to execution of the associated exception handling routine are explained below.

### 13.5.1 Undefined Instruction Exception

1. The value of the processor status word (PSW) is saved on the stack (ISP).
2. The processor mode select bit (PM), the stack pointer select bit (U), and the interrupt enable bit (I) in PSW are cleared to 0.
3. The value of the program counter (PC) is saved on the stack (ISP).
4. The vector is fetched from address FFFF FFDCh.
5. The fetched vector is set to the PC and processing branches to the exception handling routine.

### 13.5.2 Privileged Instruction Exception

1. The value in the processor status word (PSW) is saved on the stack (ISP).
2. The processor mode select bit (PM), the stack pointer select bit (U), and the interrupt enable bit (I) in PSW are cleared to 0.
3. The value of the program counter (PC) is saved on the stack (ISP).
4. The vector is fetched from address FFFF FFD0h.
5. The fetched vector is set to the PC and processing branches to the exception handling routine.

### 13.5.3 Reset

1. The control registers are initialized.
2. The vector is fetched from address FFFF FFFCh.
3. The fetched vector is set to the PC.

### 13.5.4 Non-Maskable Interrupt

1. The value of the processor status word (PSW) is saved on the stack (ISP).
2. The processor mode select bit (PM), the stack pointer select bit (U), and the interrupt enable bit (I) in PSW are cleared to 0.
3. If the interrupt was generated during the execution of an RMPA, SCMPU, SMOVB, SMOVF, SMOVU, SSTR, SUNTIL, or SWHILE instruction, the value of the program counter (PC) for that instruction is saved on the stack (ISP). For other instructions, the PC value of the next instruction is saved.
4. The processor interrupt priority level bits (IPL[3:0]) in PSW are set to Fh.
5. The vector is fetched from address FFFF FFF8h.
6. The fetched vector is set to the PC and processing branches to the exception handling routine.

### 13.5.5 Interrupt

1. The value of the processor status word (PSW) is saved on the stack (ISP) or, for the fast interrupt, in the backup PSW (BPSW).
2. The processor mode select bit (PM), the stack pointer select bit (U), and the interrupt enable bit (I) in PSW are cleared to 0.
3. If the interrupt was generated during the execution of an RMPA, SCMPU, SMOVB, SMOVF, SMOVU, SSTR, SUNTIL, or SWHILE instruction, the value of the program counter (PC) for that instruction is saved. For other instructions, the PC value of the next instruction is saved. Saving of the PC is in the backup PC (BPC) for fast interrupts.
4. The processor interrupt priority level bits (IPL[3:0]) in PSW indicate the interrupt priority level of the interrupt.
5. The vector for an interrupt source other than the fast interrupt is fetched from the relocatable vector table. For the fast interrupt, the address is fetched from the fast interrupt vector register (FINTV).
6. The fetched vector is set to the PC and processing branches to the exception handling routine.

### 13.5.6 Unconditional Trap

1. The value in the processor status word (PSW) is saved on the stack (ISP).
2. The processor mode select bit (PM), the stack pointer select bit (U), and the interrupt enable bit (I) in PSW are cleared to 0.
3. The value of the program counter (PC) for the next instruction is saved on the stack (ISP).
4. For the INT instruction, the value at the vector corresponding to the INT instruction number is fetched from the relocatable vector table.
   For the BRK instruction, the value at the vector from the start address is fetched from the relocatable vector table.
5. The fetched vector is set to the PC and processing branches to the exception handling routine.

## 13.6 Return from Exception Handling Routine

Executing the instruction listed in Table 13.3 at the end of the corresponding exception handling routine restores the values of the program counter (PC) and processor status word (PSW) that were saved on the stack or in the control registers (BPC and BPSW) immediately before the exception handling sequence.

**Table 13.3 Return from Exception Handling Routine**

| Exception | | Instruction for Return |
|---|---|---|
| Undefined instruction exception | | RTE |
| Privileged instruction exception | | RTE |
| Reset | | Return is impossible |
| Non-maskable interrupt | | Return is impossible |
| Interrupt | Fast interrupt | RTFI |
| | Other than above | RTE |
| Unconditional trap | | RTE |

## 13.7 Priority of Exception Events

The priority of exception events is listed in Table 13.4. When multiple exceptions are generated at the same time, the exception with the highest priority is accepted first.

**Table 13.4 Priority of Exception Events**

| Priority | | Exception Event |
|---|---|---|
| High ↑ | 1 | Reset |
| | 2 | Non-maskable interrupt |
| | 3 | Interrupt |
| | 4 | Undefined instruction exception<br>Privileged instruction exception |
| Low | 5 | Unconditional trap |

# 14. Interrupt Controller (ICUb)

## 14.1 Overview

The interrupt controller receives interrupt signals from peripheral modules and external pins, sends interrupts to the CPU, and activates the DTC.

Table 14.1 lists the specifications of the interrupt controller, and Figure 14.1 shows a block diagram of the interrupt controller.

**Table 14.1 Specifications of Interrupt Controller**

| Item | | Description |
|---|---|---|
| Interrupts | Peripheral function interrupts | • Interrupts from peripheral modules<br>• Interrupt detection: Edge detection/level detection<br>Edge detection or level detection is fixed for each source of connected peripheral modules. |
| | External pin interrupts | • Interrupts from pins IRQ0 to IRQ7<br>• Number of sources: 8<br>• Interrupt detection: Low/falling edge/rising edge/rising and falling edges<br>One of these detection methods can be set for each source.<br>• Digital filter function: Supported |
| | Software interrupt | • Interrupt generated by writing to a register<br>• One interrupt source |
| | Event link interrupt | The ELSR18I interrupt is generated by an ELC event |
| | Interrupt priority | Specified by registers. |
| | Fast interrupt function | Faster interrupt handling of the CPU can be set only for a single interrupt source. |
| | DTC control | The DTC can be activated by interrupt sources.*1 |
| Non-maskable interrupts | NMI pin interrupt | • Interrupt from the NMI pin<br>• Interrupt detection: Falling edge/rising edge<br>• Digital filter function: Supported |
| | Oscillation stop detection interrupt | Interrupt on detection of oscillation having stopped |
| | IWDT underflow/ refresh error | Interrupt on an underflow of the down counter or occurrence of a refresh error |
| | Voltage monitoring 1 interrupt | Voltage monitoring interrupt of voltage monitoring circuit 1 (LVD1) |
| | Voltage monitoring 2 interrupt | Voltage monitoring interrupt of voltage monitoring circuit 2 (LVD2) |
| Return from power-down modes | | • Sleep mode, deep sleep mode: Return is initiated by non-maskable interrupts or any other interrupt source.<br>• Software standby mode: Return is initiated by non-maskable interrupts, IRQ0 to IRQ7 interrupts, or RTC alarm/periodic interrupts. |

Note 1. For the DTC activation source, see Table 14.3, Interrupt Vector Table.

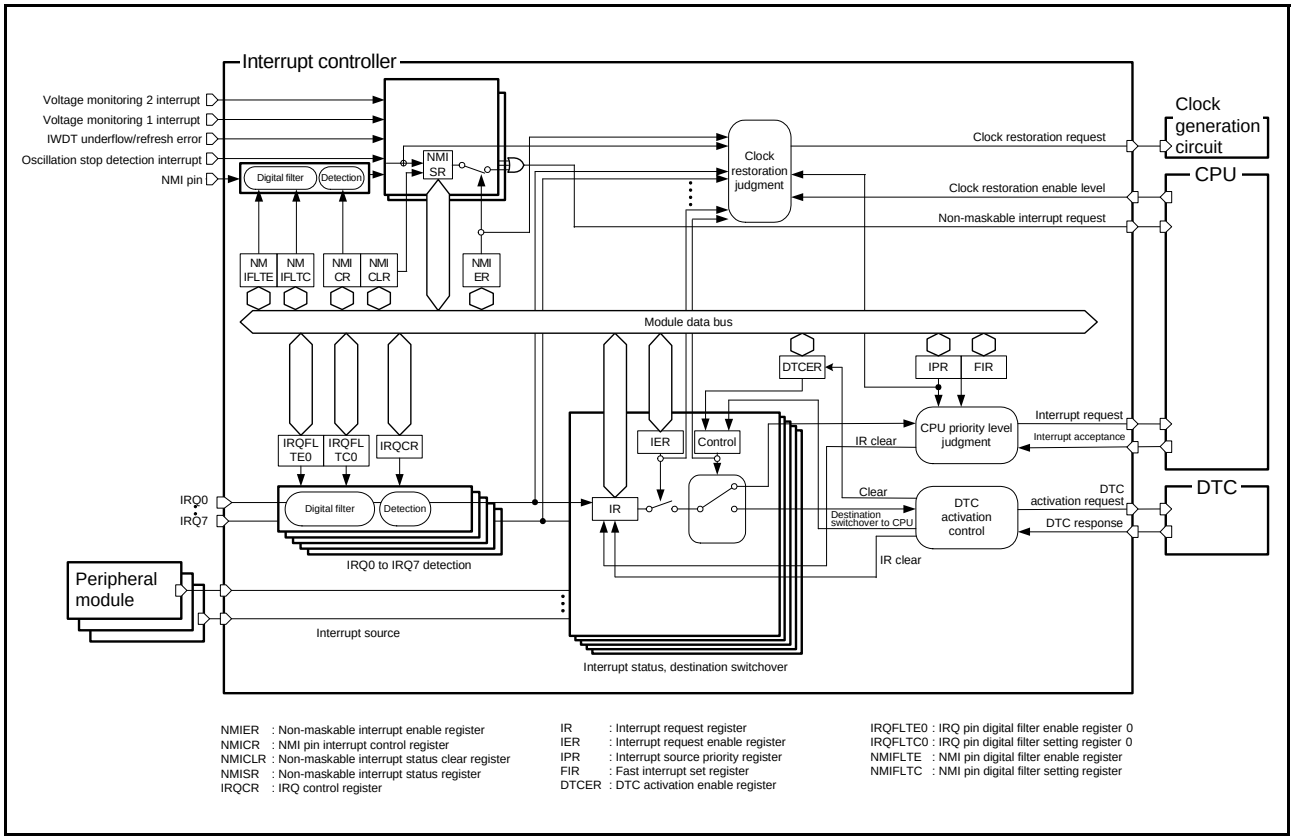



**Figure 14.1 Block Diagram of Interrupt Controller**

Table 14.2 lists the I/O pins of the interrupt controller.

**Table 14.2 Pin Configuration of Interrupt Controller**

| Pin Name | I/O | Description |
|---|---|---|
| NMI | Input | Non-maskable interrupt request pin |
| IRQ0 to IRQ7 | Input | External interrupt request pins |

## 14.2 Register Descriptions

### 14.2.1 Interrupt Request Register n (IRn) (n = interrupt vector number)

Address(es): 0008 7010h to 0008 70F9h

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | IR |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b0 | IR | Interrupt Status Flag | 0: Interrupt not requested<br>1: Interrupt requested | R/(W) *1 |
| b7 to b1 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |

Note 1. For an edge detection interrupt, only 0 can be written to this bit; do not write 1.
For a level detection interrupt, neither 0 nor 1 can be written.

IRn is provided for each interrupt source, where “n” indicates the interrupt vector number.
For the correspondence between interrupt sources and interrupt vector numbers, see Table 14.3, Interrupt Vector Table.

**IR Flag (Interrupt Status Flag)**

This bit is the status flag of an individual interrupt request. This flag is set to 1 when the corresponding interrupt request is generated. To detect an interrupt request, the interrupt request output should be enabled by the corresponding peripheral module interrupt enable bit.
There are two interrupt request detection methods: edge detection and level detection. For interrupts from peripheral modules, either edge detection or level detection is determined for each interrupt source. For interrupts from IRQi pins, edge detection or level detection is selected by setting the corresponding IRQCRi.IRQMD[1:0] bits (i = 0 to 7). For detection of the various interrupt sources, see Table 14.3, Interrupt Vector Table.

#### (1) Edge detection

[Setting condition]

- The flag is set to 1 in response to the generation of an interrupt request from the corresponding peripheral module or IRQi pin. For interrupt generation by the various peripheral modules, refer to the sections describing the modules.

[Clearing conditions]

- The flag is cleared to 0 when the interrupt request destination accepts the interrupt request.
- The IR flag is cleared to 0 by writing 0 to it. Note, however, that writing 0 to the IR flag is prohibited if the destination of the interrupt request is the DTC.

#### (2) Level detection

[Setting condition]

- The flag remains set to 1 while an interrupt request is being sent from the corresponding peripheral module or IRQi pin. For interrupt generation by the various peripheral modules, refer to the sections describing the modules.

[Clearing condition]

- The flag is cleared to 0 when the source of the interrupt request is cleared (it is not cleared when the interrupt request destination accepts the interrupt request). For clearing interrupts from the various peripheral modules, refer to the sections describing the modules.

When level detection has been selected for an IRQi pin, the interrupt request is withdrawn by driving the IRQi pin high.
Do not write 0 or 1 to the IR flag while level detection is selected.

## 14.2.2 Interrupt Request Enable Register m (IERm) (m = 02h to 1Fh)

Address(es): 0008 7202h to 0008 721Fh

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | IEN7 | IEN6 | IEN5 | IEN4 | IEN3 | IEN2 | IEN1 | IEN0 |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b0 | IEN0 | Interrupt Request Enable 0 | 0: Interrupt request is disabled<br>1: Interrupt request is enabled | R/W |
| b1 | IEN1 | Interrupt Request Enable 1 | | R/W |
| b2 | IEN2 | Interrupt Request Enable 2 | | R/W |
| b3 | IEN3 | Interrupt Request Enable 3 | | R/W |
| b4 | IEN4 | Interrupt Request Enable 4 | | R/W |
| b5 | IEN5 | Interrupt Request Enable 5 | | R/W |
| b6 | IEN6 | Interrupt Request Enable 6 | | R/W |
| b7 | IEN7 | Interrupt Request Enable 7 | | R/W |

Note: • Write 0 to the bit that corresponds to the vector number for reservation. These bits are read as 0.

**IENj Bit (Interrupt Request Enable j) (j = 0 to 7)**

When an IENj bit is 1, the corresponding interrupt request will be output to the destination selected for the request.
When an IENj bit is 0, the corresponding interrupt request will not be output to the destination selected for the request.
The setting of an IENj bit does not affect the IRn.IR flag. Even if the corresponding IENj bit is 0, the IR flag value changes according to the descriptions in section 14.2.1, Interrupt Request Register n (IRn) (n = interrupt vector number).
The IERm.IENj bit is set for each request source (vector number).
For the correspondence between interrupt sources and IERm.IENj bit, see Table 14.3, Interrupt Vector Table.
For the procedure for setting IERm.IENj bit during the selection of destinations for interrupt requests, refer to section 14.4.3, Selecting Interrupt Request Destinations.

## 14.2.3 Interrupt Source Priority Register n (IPRn) (n = 000 to 249)

Address(es): 0008 7300h to 0008 73F9h

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | IPR[3:0] | | | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b3 to b0 | IPR[3:0] | Interrupt Priority Level Select | b3 b0<br>0 0 0 0: Level 0 (interrupt disabled)*1<br>0 0 0 1: Level 1<br>0 0 1 0: Level 2<br>0 0 1 1: Level 3<br>0 1 0 0: Level 4<br>0 1 0 1: Level 5<br>0 1 1 0: Level 6<br>0 1 1 1: Level 7<br>1 0 0 0: Level 8<br>1 0 0 1: Level 9<br>1 0 1 0: Level 10<br>1 0 1 1: Level 11<br>1 1 0 0: Level 12<br>1 1 0 1: Level 13<br>1 1 1 0: Level 14<br>1 1 1 1: Level 15 (highest) | R/W |
| b7 to b4 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |

Note 1. When the interrupt is specified as a fast interrupt, it can be issued even if the priority level is level 0.

For the correspondence between interrupt sources and IPRn register, see Table 14.3, Interrupt Vector Table.

**IPR[3:0] Bits (Interrupt Priority Level Select)**

These bits specify the priority level of the corresponding interrupt source.

Priority levels specified by the IPR[3:0] bits are used only to determine the priority of interrupt requests to be transferred to the CPU, and do not affect activation requests to the DTC.

The CPU accepts only interrupt requests higher than the priority level specified by the PSW.IPL[3:0] bits, and handles accepted interrupts.

If two or more interrupt requests are generated at the same time, their priority levels are compared with the value of the IPR[3:0] bits. If interrupt requests of the same priority level are generated at the same time, an interrupt source with a smaller vector number takes precedence.

These bits should be written to while an interrupt request is disabled (IERm.IENj bit = 0).

## 14.2.4 Fast Interrupt Set Register (FIR)

Address(es): 0008 72F0h

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | FIEN | — | — | — | — | — | — | — | FVCT[7:0] | | | | | | | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b7 to b0 | FVCT[7:0] | Fast Interrupt Vector Number | Specify the vector number of an interrupt source to be a fast interrupt. | R/W |
| b14 to b8 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |
| b15 | FIEN | Fast Interrupt Enable | 0: Fast interrupt is disabled<br>1: Fast interrupt is enabled | R/W |

The fast interrupt function based on the FIR register setting is applicable only to interrupts to the CPU. It will not affect any transfer request to the DTC.
Before writing to this register, be sure to disable interrupt requests (IERm.IENj bit = 0).

### FVCT[7:0] Bits (Fast Interrupt Vector Number)

The FVCT[7:0] bits specify the vector number of an interrupt source that uses the fast interrupt function.

### FIEN Bit (Fast Interrupt Enable)

This bit enables the fast interrupt.
Setting this bit to 1 makes the interrupt request of the vector number specified by the FVCT[7:0] bits a fast interrupt. When an interrupt request of the vector number specified by the FVCT[7:0] bits is generated and the interrupt request destination is the CPU while the FIEN bit is 1, the interrupt request is output to the CPU as a fast interrupt regardless of the setting of the IPRn register. When using the fast interrupt for returning from the software standby mode, refer to section 14.6.2, Return from Software Standby Mode.
If the setting of the IERm.IENj (m = 02h to 1Fh, j = 0 to 7) bit has disabled interrupt requests from the interrupt source with the vector number in this register, fast interrupt requests are not output to the CPU.
For settable vector numbers, see Table 14.3, Interrupt Vector Table.
Do not write any reserved vector numbers to the FVCT[7:0] bits.

For details on the fast interrupt, refer to section 13, Exception Handling, and section 14.4.5, Fast Interrupt.

## 14.2.5 Software Interrupt Activation Register (SWINTR)

Address(es): 0008 72E0h

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | SWINT |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b0 | SWINT | Software Interrupt Activation | This bit is read as 0. Writing 1 issues a software interrupt request. Writing 0 to this bit has no effect. | R/(W) *1 |
| b7 to b1 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |

Note 1. Only 1 can be written.

**SWINT Bit (Software Interrupt Activation)**

When 1 is written to the SWINT bit, interrupt request register 027 (IR027) is set to 1.

If 1 is written to the SWINT bit when DTC activation enable register 027 (DTCER027) is set to 0, an interrupt to the CPU is generated.

If 1 is written to the SWINT bit when DTC activation enable register 027 (DTCER027) is set to 1, a DTC activation request is issued.

## 14.2.6 DTC Activation Enable Register n (DTCERn) (n = interrupt vector number)

Address(es): 0008 711Bh to 0008 71F8h

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | DTCE |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b0 | DTCE | DTC Activation Enable | 0: DTC activation is disabled<br>1: DTC activation is enabled | R/W |
| b7 to b1 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |

See Table 14.3, Interrupt Vector Table, for the interrupt sources that are selectable as sources for DTC activation.

**DTCE Bit (DTC Activation Enable)**

When the DTCE bit is set to 1, the corresponding interrupt source is selected as the source for the DTC activation.

[Setting condition]

- When 1 is written to the DTCE bit

[Clearing conditions]

- When the specified number of transfers is completed (for the chain transfer, the number of transfers for the last chain transfer is completed)
- When 0 is written to the DTCE bit

## 14.2.7 IRQ Control Register i (IRQCRi) (i = 0 to 7)

Address(es): 0008 7500h to 0008 7507h

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | IRQMD[1:0] | | — | — |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b1, b0 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |
| b3, b2 | IRQMD[1:0] | IRQ Detection Sense Select | b3 b2<br>0 0: Low level<br>0 1: Falling edge<br>1 0: Rising edge<br>1 1: Rising and falling edges | R/W |
| b7 to b4 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |

Only change the settings of this register while the corresponding interrupt request enable bit is prohibiting the interrupt request (IERm.IENj bit is 0). After changing the setting, clear the IR flag before setting the interrupt enable bit. However, when the change is to the low level, the IR flag does not require clearing.

**IRQMD[1:0] Bits (IRQ Detection Sense Select)**

These bits select the detection sensing method of external pin interrupt sources IRQ0 to IRQ7.
For the external pin interrupt detection setting, refer to section 14.4.7, External Pin Interrupts.

## 14.2.8 IRQ Pin Digital Filter Enable Register 0 (IRQFLTE0)

Address(es): 0008 7510h

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | FLTEN7 | FLTEN6 | FLTEN5 | FLTEN4 | FLTEN3 | FLTEN2 | FLTEN1 | FLTEN0 |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b0 | FLTEN0 | IRQ0 Digital Filter Enable | 0: Digital filter disabled<br>1: Digital filter enabled | R/W |
| b1 | FLTEN1 | IRQ1 Digital Filter Enable | | R/W |
| b2 | FLTEN2 | IRQ2 Digital Filter Enable | | R/W |
| b3 | FLTEN3 | IRQ3 Digital Filter Enable | | R/W |
| b4 | FLTEN4 | IRQ4 Digital Filter Enable | | R/W |
| b5 | FLTEN5 | IRQ5 Digital Filter Enable | | R/W |
| b6 | FLTEN6 | IRQ6 Digital Filter Enable | | R/W |
| b7 | FLTEN7 | IRQ7 Digital Filter Enable | | R/W |

**FLTENi Bit (IRQi Digital Filter Enable) (i = 0 to 7)**

These bits enable the digital filter used for the external pin interrupt sources IRQ0 to IRQ7.

The digital filter is enabled when the FLTENi bit is 1, and disabled when the FLTENi bit is 0.

The IRQi pin level is sampled at the sampling clock cycle specified with the IRQFLTC0.FCLKSELi[1:0] bits. When the sampled level matches three times, the output level from the digital filter changes.

For details of the digital filter, refer to section 14.4.6, Digital Filter.

## 14.2.9 IRQ Pin Digital Filter Setting Register 0 (IRQFLTC0)

Address(es): 0008 7514h

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | FCLKSEL7[1:0] | | FCLKSEL6[1:0] | | FCLKSEL5[1:0] | | FCLKSEL4[1:0] | | FCLKSEL3[1:0] | | FCLKSEL2[1:0] | | FCLKSEL1[1:0] | | FCLKSEL0[1:0] | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b1, b0 | FCLKSEL0[1:0] | IRQ0 Digital Filter Sampling Clock | 0 0: PCLK<br>0 1: PCLK/8<br>1 0: PCLK/32<br>1 1: PCLK/64 | R/W |
| b3, b2 | FCLKSEL1[1:0] | IRQ1 Digital Filter Sampling Clock | | R/W |
| b5, b4 | FCLKSEL2[1:0] | IRQ2 Digital Filter Sampling Clock | | R/W |
| b7, b6 | FCLKSEL3[1:0] | IRQ3 Digital Filter Sampling Clock | | R/W |
| b9, b8 | FCLKSEL4[1:0] | IRQ4 Digital Filter Sampling Clock | | R/W |
| b11, b10 | FCLKSEL5[1:0] | IRQ5 Digital Filter Sampling Clock | | R/W |
| b13, b12 | FCLKSEL6[1:0] | IRQ6 Digital Filter Sampling Clock | | R/W |
| b15, b14 | FCLKSEL7[1:0] | IRQ7 Digital Filter Sampling Clock | | R/W |

**FCLKSELi[1:0] Bits (IRQi Digital Filter Sampling Clock) (i = 0 to 7)**

These bits select the cycle of the digital filter sampling clock for the external pin interrupt request pins IRQ0 to IRQ7. The sampling clock cycle can be selected from among the PCLK (every cycle), PCLK/8 (once every eight cycles), PCLK/32 (once every 32 cycles), and PCLK/64 (once every 64 cycles).

For details of the digital filter, refer to section 14.4.6, Digital Filter.

## 14.2.10 Non-Maskable Interrupt Status Register (NMISR)

Address(es): 0008 7580h

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | LVD2ST | LVD1ST | IWDTST | — | OSTST | NMIST |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b0 | NMIST | NMI Status Flag | 0: NMI pin interrupt is not requested<br>1: NMI pin interrupt is requested | R |
| b1 | OSTST | Oscillation Stop Detection Interrupt Status Flag | 0: Oscillation stop detection interrupt is not requested<br>1: Oscillation stop detection interrupt is requested | R |
| b2 | — | Reserved | This bit is read as 0 and cannot be modified. | R |
| b3 | IWDTST | IWDT Underflow/Refresh Error Status Flag | 0: IWDT underflow/refresh error interrupt is not requested<br>1: IWDT underflow/refresh error interrupt is requested | R |
| b4 | LVD1ST | Voltage Monitoring 1 Interrupt Status Flag | 0: Voltage monitoring 1 interrupt is not requested<br>1: Voltage monitoring 1 interrupt is requested | R |
| b5 | LVD2ST | Voltage Monitoring 2 Interrupt Status Flag | 0: Voltage monitoring 2 interrupt is not requested<br>1: Voltage monitoring 2 interrupt is requested | R |
| b7, b6 | — | Reserved | These bits are read as 0 and cannot be modified. | R |

The NMISR register monitors the status of a non-maskable interrupt source. Writing to the NMISR register is ignored. The setting in the non-maskable interrupt enable register (NMIER) does not affect the status flags in NMISR.
Before the end of the non-maskable interrupt handler, read the NMISR register and confirm the generation status of other non-maskable interrupts. Be sure to confirm that all of the bits in the NMISR register are set to 0 before the end of the handler.

### NMIST Flag (NMI Status Flag)

This flag indicates the NMI pin interrupt request.
The NMIST flag is read-only, and cleared by the NMICLR.NMICLR bit.
[Setting condition]

- When an edge specified by the NMICR.NMIMD bit is input to the NMI pin

[Clearing condition]

- When 1 is written to the NMICLR.NMICLR bit

### OSTST Flag (Oscillation Stop Detection Interrupt Status Flag)

This flag indicates the oscillation stop detection interrupt request.
The OSTST flag is read-only, and cleared by the NMICLR.OSTCLR bit.
[Setting condition]

- When the oscillation stop detection interrupt is generated

[Clearing condition]

- When 1 is written to the NMICLR.OSTCLR bit

### IWDTST Flag (IWDT Underflow/Refresh Error Status Flag)

This flag indicates the IWDT underflow/refresh error interrupt request.
The IWDTST flag is read-only, and cleared by the NMICLR.IWDTCLR bit.
[Setting condition]

- When the IWDT underflow/refresh error interrupt is generated while this interrupt is enabled at its source.

[Clearing condition]

- When 1 is written to the NMICLR.IWDTCLR bit

### LVD1ST Flag (Voltage Monitoring 1 Interrupt Status Flag)

This flag indicates the request for voltage monitoring 1 interrupt.
The LVD1ST flag is read-only, and cleared by the NMICLR.LVD1CLR bit.
[Setting condition]

- When the voltage monitoring 1 interrupt is generated while this interrupt is enabled at its source.

[Clearing condition]

- When 1 is written to the NMICLR.LVD1CLR bit

### LVD2ST Flag (Voltage Monitoring 2 Interrupt Status Flag)

This flag indicates the request for voltage monitoring 2 interrupt.
The LVD2ST flag is read-only, and cleared by the NMICLR.LVD2CLR bit.
[Setting condition]

- When the voltage monitoring 2 interrupt is generated while this interrupt is enabled at its source.

[Clearing condition]

- When 1 is written to the NMICLR.LVD2CLR bit

## 14.2.11 Non-Maskable Interrupt Enable Register (NMIER)

Address(es): 0008 7581h

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | LVD2EN | LVD1EN | IWDTEN | — | OSTEN | NMIEN |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b0 | NMIEN | NMI Pin Interrupt Enable | 0: NMI pin interrupt is disabled<br>1: NMI pin interrupt is enabled | R/(W) *1 |
| b1 | OSTEN | Oscillation Stop Detection Interrupt Enable | 0: Oscillation stop detection interrupt is disabled<br>1: Oscillation stop detection interrupt is enabled | R/(W) *1 |
| b2 | — | Reserved | This bit is read as 0. The write value should be 0. | R/(W) |
| b3 | IWDTEN | IWDT Underflow/Refresh Error Enable | 0: IWDT underflow/refresh error interrupt is disabled<br>1: IWDT underflow/refresh error interrupt is enabled | R/(W) *1 |
| b4 | LVD1EN | Voltage Monitoring 1 Interrupt Enable | 0: Voltage monitoring 1 interrupt is disabled<br>1: Voltage monitoring 1 interrupt is enabled | R/(W) *1 |
| b5 | LVD2EN | Voltage Monitoring 2 Interrupt Enable | 0: Voltage monitoring 2 interrupt is disabled<br>1: Voltage monitoring 2 interrupt is enabled | R/(W) *1 |
| b7, b6 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |

Note 1. 1 can be written to this bit only once, and subsequent write accesses are no longer enabled.

### NMIEN Bit (NMI Pin Interrupt Enable)

This bit enables the NMI pin interrupt.

A 1 can be written to this bit only once, and subsequent write accesses are no longer enabled.

Writing 0 to this bit is disabled.

### OSTEN Bit (Oscillation Stop Detection Interrupt Enable)

This bit enables the oscillation stop detection interrupt.

A 1 can be written to this bit only once, and subsequent write accesses are no longer enabled.

Writing 0 to this bit is disabled.

### IWDTEN Bit (IWDT Underflow/Refresh Error Enable)

This bit enables the IWDT underflow/refresh error interrupt.

A 1 can be written to this bit only once, and subsequent write accesses are no longer enabled.

Writing 0 to this bit is disabled.

### LVD1EN Bit (Voltage Monitoring 1 Interrupt Enable)

This bit enables the voltage monitoring 1 interrupt.

A 1 can be written to this bit only once, and subsequent write accesses are no longer enabled.

Writing 0 to this bit is disabled.

### LVD2EN Bit (Voltage Monitoring 2 Interrupt Enable)

This bit enables the voltage monitoring 2 interrupt.

A 1 can be written to this bit only once, and subsequent write accesses are no longer enabled.

Writing 0 to this bit is disabled.

## 14.2.12 Non-Maskable Interrupt Status Clear Register (NMICLR)

Address(es): 0008 7582h

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | LVD2CLR | LVD1CLR | IWDTCLR | — | OSTCLR | NMICLR |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b0 | NMICLR | NMI Clear | This bit is read as 0. Writing 1 to this bit clears the NMISR.NMIST flag. Writing 0 to this bit has no effect. | R/(W) *1 |
| b1 | OSTCLR | OST Clear | This bit is read as 0. Writing 1 to this bit clears the NMISR.OSTST flag. Writing 0 to this bit has no effect. | R/(W) *1 |
| b2 | — | Reserved | This bit is read as 0. The write value should be 0. | R/(W) |
| b3 | IWDTCLR | IWDT Clear | This bit is read as 0. Writing 1 to this bit clears the NMISR.IWDTST flag. Writing 0 to this bit has no effect. | R/(W) *1 |
| b4 | LVD1CLR | LVD1 Clear | This bit is read as 0. Writing 1 to this bit clears the NMISR.LVD1ST flag. Writing 0 to this bit has no effect. | R/(W) *1 |
| b5 | LVD2CLR | LVD2 Clear | This bit is read as 0. Writing 1 to this bit clears the NMISR.LVD2ST flag. Writing 0 to this bit has no effect. | R/(W) *1 |
| b7, b6 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |

Note 1. Only 1 can be written to this bit.

### NMICLR Bit (NMI Clear)

Writing 1 to the NMICLR bit clears the NMISR.NMIST flag. This bit is read as 0.

### OSTCLR Bit (OST Clear)

Writing 1 to the OSTCLR bit clears the NMISR.OSTST flag. This bit is read as 0.

### IWDTCLR Bit (IWDT Clear)

Writing 1 to the IWDTCLR bit clears the NMISR.IWDTST flag. This bit is read as 0.

### LVD1CLR Bit (LVD1 Clear)

Writing 1 to the LVD1CLR bit clears the NMISR.LVD1ST flag. This bit is read as 0.

### LVD2CLR Bit (LVD2 Clear)

Writing 1 to the LVD2CLR bit clears the NMISR.LVD2ST flag. This bit is read as 0.

## 14.2.13 NMI Pin Interrupt Control Register (NMICR)

Address(es): 0008 7583h

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | NMIMD | — | — | — |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b2 to b0 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |
| b3 | NMIMD | NMI Detection Set | 0: Falling edge<br>1: Rising edge | R/W |
| b7 to b4 | — | Reserved | These bits are read as 0 and cannot be modified. | R/W |

Change the setting of the NMICR register before the NMI pin interrupt is enabled (before setting the NMIER.NMIEN bit to 1).

**NMIMD Bit (NMI Detection Set)**

This bit specifies the detection edge of the NMI pin interrupt.

## 14.2.14 NMI Pin Digital Filter Enable Register (NMIFLTE)

Address(es): 0008 7590h

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | NFLTEN |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b0 | NFLTEN | NMI Digital Filter Enable | 0: Digital filter disabled.<br>1: Digital filter enabled. | R/W |
| b7 to b1 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |

**NFLTEN Bit (NMI Digital Filter Enable)**

This bit enables the digital filter used for the NMI pin interrupt.

The digital filter is enabled when the NFLTEN bit is 1, and disabled when the NFLTEN bit is 0.

The NMI pin level is sampled at the sampling clock cycle specified with the NMIFLTC.NFCLKSEL[1:0] bits. When the sampled level matches three times, the output level from the digital filter changes.

For details of the digital filter, refer to section 14.4.6, Digital Filter.

## 14.2.15 NMI Pin Digital Filter Setting Register (NMIFLTC)

Address(es): 0008 7594h

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | NFCLKSEL[1:0] | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b1, b0 | NFCLKSEL[1:0] | NMI Digital Filter Sampling Clock | b1 b0<br>0 0: PCLK<br>0 1: PCLK/8<br>1 0: PCLK/32<br>1 1: PCLK/64 | R/W |
| b7 to b2 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |

**NFCLKSEL[1:0] Bits (NMI Digital Filter Sampling Clock)**

These bits select the cycle of the digital filter sampling clock for the NMI pin interrupt.

The sampling clock cycle can be selected from among the PCLK (every cycle), PCLK/8 (once every eight cycles), PCLK/32 (once every 32 cycles), and PCLK/64 (once every 64 cycles).

For details of the digital filter, refer to section 14.4.6, Digital Filter.

## 14.3 Vector Table

There are two types of interrupts detected by the interrupt controller: maskable interrupts and non-maskable interrupts. When the CPU accepts an interrupt or non-maskable interrupt, it acquires a 4-byte vector address from the vector table.

### 14.3.1 Interrupt Vector Table

The interrupt vector table is placed in the 1024-byte range (4 bytes × 256 sources) beginning at the address specified in the interrupt table register (INTB) of the CPU. Write a value to the INTB register before enabling interrupts. The value written to the INTB register should be a multiple of 4.
Executing an INT instruction or BRK instruction leads to the generation of an unconditional trap. The same range of memory as shown in Table 14.3, Interrupt Vector Table, is used for the vectors for unconditional traps. The vector for BRK instructions is vector 0 while the vector numbers for INT instructions are specifiable as numbers in the range from 0 to 255.
Table 14.3 lists details of the interrupt vectors. Details of the headings in Table 14.3 are listed below.

| Item | Description |
|---|---|
| Source of interrupt request generation | Name of the source for generation of the interrupt request |
| Name | Name of the interrupt |
| Vector no. | Vector number for the interrupt |
| Vector address offset | Value of the offset from the base address for the vector table |
| Form of interrupt detection | “Edge” or “level” as the method for detection of the interrupt |
| CPU interrupt | “○” in this column indicates usability as a CPU interrupt. |
| DTC activation | “○” in this column indicates usability as a request for DTC activation. |
| sstb return | “○” in this column indicates usability as a request for return from software standby mode. |
| IER | Name of the interrupt request enable register (IER) and bit corresponding to the vector number |
| IPR | Name of the interrupt source priority register (IPR) corresponding to the interrupt source |
| DTCER | Name of the DTC activation enable register (DTCER) corresponding to the DTC activation source |

**Table 14.3 Interrupt Vector Table (1/7)**

| Source of Interrupt Request Generation | Name | Vector No.*1 | Vector Address Offset | Form of Interrupt Detection | CPU | DTC | sstb Return | IER | IPR | DTCER |
|---|---|---|---|---|---|---|---|---|---|---|
| — | For an unconditional trap | 0 | 0000h | — | × | × | × | — | — | — |
| — | For an unconditional trap | 1 | 0004h | — | × | × | × | — | — | — |
| — | For an unconditional trap | 2 | 0008h | — | × | × | × | — | — | — |
| — | For an unconditional trap | 3 | 000Ch | — | × | × | × | — | — | — |
| — | For an unconditional trap | 4 | 0010h | — | × | × | × | — | — | — |
| — | For an unconditional trap | 5 | 0014h | — | × | × | × | — | — | — |
| — | For an unconditional trap | 6 | 0018h | — | × | × | × | — | — | — |
| — | For an unconditional trap | 7 | 001Ch | — | × | × | × | — | — | — |
| — | For an unconditional trap | 8 | 0020h | — | × | × | × | — | — | — |
| — | For an unconditional trap | 9 | 0024h | — | × | × | × | — | — | — |
| — | For an unconditional trap | 10 | 0028h | — | × | × | × | — | — | — |
| — | For an unconditional trap | 11 | 002Ch | — | × | × | × | — | — | — |
| — | For an unconditional trap | 12 | 0030h | — | × | × | × | — | — | — |
| — | For an unconditional trap | 13 | 0034h | — | × | × | × | — | — | — |
| — | For an unconditional trap | 14 | 0038h | — | × | × | × | — | — | — |
| — | For an unconditional trap | 15 | 003Ch | — | × | × | × | — | — | — |
| BSC | BUSERR | 16 | 0040h | Level | ○ | × | × | IER02.IEN0 | IPR000 | — |
| — | Reserved | 17 | 0044h | — | × | × | × | — | — | — |
| — | Reserved | 18 | 0048h | — | × | × | × | — | — | — |
| — | Reserved | 19 | 004Ch | — | × | × | × | — | — | — |
| — | Reserved | 20 | 0050h | — | × | × | × | — | — | — |

**Table 14.3 Interrupt Vector Table (2/7)**

| Source of Interrupt Request Generation | Name | Vector No.*1 | Vector Address Offset | Form of Interrupt Detection | CPU | DTC | sstb Return | IER | IPR | DTCER |
|---|---|---|---|---|---|---|---|---|---|---|
| — | Reserved | 21 | 0054h | — | × | × | × | — | — | — |
| — | Reserved | 22 | 0058h | — | × | × | × | — | — | — |
| — | Reserved | 23 | 005Ch | — | × | × | × | — | — | — |
| — | Reserved | 24 | 0060h | — | × | × | × | — | — | — |
| — | Reserved | 25 | 0064h | — | × | × | × | — | — | — |
| — | Reserved | 26 | 0068h | — | × | × | × | — | — | — |
| ICU | SWINT | 27 | 006Ch | Edge | ○ | ○ | × | IER03.IEN3 | IPR003 | DTCER027 |
| CMT0 | CMI0 | 28 | 0070h | Edge | ○ | ○ | × | IER03.IEN4 | IPR004 | DTCER028 |
| CMT1 | CMI1 | 29 | 0074h | Edge | ○ | ○ | × | IER03.IEN5 | IPR005 | DTCER029 |
| — | Reserved | 30 | 0078h | — | × | × | × | — | — | — |
| — | Reserved | 31 | 007Ch | — | × | × | × | — | — | — |
| CAC | FERRF | 32 | 0080h | Level | ○ | × | × | IER04.IEN0 | IPR032 | — |
| | MENDF | 33 | 0084h | Level | ○ | × | × | IER04.IEN1 | IPR033 | — |
| | OVFF | 34 | 0088h | Level | ○ | × | × | IER04.IEN2 | IPR034 | — |
| — | Reserved | 35 | 008Ch | — | × | × | × | — | — | — |
| USB0 | D0FIFO0 | 36 | 0090h | Edge | ○ | ○ | × | IER04.IEN4 | IPR036 | DTCER036 |
| | D1FIFO0 | 37 | 0094h | Edge | ○ | ○ | × | IER04.IEN5 | IPR037 | DTCER037 |
| | USBI0 | 38 | 0098h | Edge | ○ | × | × | IER04.IEN6 | IPR038 | — |
| — | Reserved | 39 | 009Ch | — | × | × | × | — | — | — |
| — | Reserved | 40 | 00A0h | — | × | × | × | — | — | — |
| — | Reserved | 41 | 00A4h | — | × | × | × | — | — | — |
| — | Reserved | 42 | 00A8h | — | × | × | × | — | — | — |
| — | Reserved | 43 | 00ACh | — | × | × | × | — | — | — |
| RSPI0 | SPEI0 | 44 | 00B0h | Level | ○ | × | × | IER05.IEN4 | IPR044 | — |
| | SPRI0 | 45 | 00B4h | Edge | ○ | ○ | × | IER05.IEN5 | | DTCER045 |
| | SPTI0 | 46 | 00B8h | Edge | ○ | ○ | × | IER05.IEN6 | | DTCER046 |
| | SPII0 | 47 | 00BCh | Level | ○ | × | × | IER05.IEN7 | | — |
| — | Reserved | 48 | 00D0h | — | × | × | × | — | — | — |
| — | Reserved | 49 | 00D4h | — | × | × | × | — | — | — |
| — | Reserved | 50 | 00D8h | — | × | × | × | — | — | — |
| — | Reserved | 51 | 00DCh | — | × | × | × | — | — | — |
| — | Reserved | 52 | 00D0h | — | × | × | × | — | — | — |
| — | Reserved | 53 | 00D4h | — | × | × | × | — | — | — |
| — | Reserved | 54 | 00D8h | — | × | × | × | — | — | — |
| — | Reserved | 55 | 00DCh | — | × | × | × | — | — | — |
| — | Reserved | 56 | 00E0h | — | × | × | × | — | — | — |
| DOC | DOPCF | 57 | 00E4h | Level | ○ | × | × | IER07.IEN1 | IPR057 | — |
| — | Reserved | 58 | 00E8h | — | × | × | × | — | — | — |
| — | Reserved | 59 | 00ECh | — | × | × | × | — | — | — |
| — | Reserved | 60 | 00F0h | — | × | × | × | — | — | — |
| — | Reserved | 61 | 00F4h | — | × | × | × | — | — | — |
| — | Reserved | 62 | 00F8h | — | × | × | × | — | — | — |
| RTC | CUP | 63 | 00FCh | Edge | ○ | × | × | IER07.IEN7 | IPR063 | — |

**Table 14.3 Interrupt Vector Table (3/7)**

| Source of Interrupt Request Generation | Name | Vector No.*1 | Vector Address Offset | Form of Interrupt Detection | CPU | DTC | sstb Return | IER | IPR | DTCER |
|---|---|---|---|---|---|---|---|---|---|---|
| ICU | IRQ0 | 64 | 0100h | Edge/Level | ○ | ○ | ○ | IER08.IEN0 | IPR064 | DTCER064 |
| | IRQ1 | 65 | 0104h | Edge/Level | ○ | ○ | ○ | IER08.IEN1 | IPR065 | DTCER065 |
| | IRQ2 | 66 | 0108h | Edge/Level | ○ | ○ | ○ | IER08.IEN2 | IPR066 | DTCER066 |
| | IRQ3 | 67 | 010Ch | Edge/Level | ○ | ○ | ○ | IER08.IEN3 | IPR067 | DTCER067 |
| | IRQ4 | 68 | 0110h | Edge/Level | ○ | ○ | ○ | IER08.IEN4 | IPR068 | DTCER068 |
| | IRQ5 | 69 | 0114h | Edge/Level | ○ | ○ | ○ | IER08.IEN5 | IPR069 | DTCER069 |
| | IRQ6 | 70 | 0118h | Edge/Level | ○ | ○ | ○ | IER08.IEN6 | IPR070 | DTCER070 |
| | IRQ7 | 71 | 011Ch | Edge/Level | ○ | ○ | ○ | IER08.IEN7 | IPR071 | DTCER071 |
| — | Reserved | 72 | 0120h | — | × | × | × | — | — | — |
| — | Reserved | 73 | 0124h | — | × | × | × | — | — | — |
| — | Reserved | 74 | 0128h | — | × | × | × | — | — | — |
| — | Reserved | 75 | 012Ch | — | × | × | × | — | — | — |
| — | Reserved | 76 | 0130h | — | × | × | × | — | — | — |
| — | Reserved | 77 | 0134h | — | × | × | × | — | — | — |
| — | Reserved | 78 | 0138h | — | × | × | × | — | — | — |
| — | Reserved | 79 | 013Ch | — | × | × | × | — | — | — |
| — | Reserved | 80 | 0140h | — | × | × | × | — | — | — |
| — | Reserved | 81 | 0144h | — | × | × | × | — | — | — |
| — | Reserved | 82 | 0148h | — | × | × | × | — | — | — |
| — | Reserved | 83 | 014Ch | — | × | × | × | — | — | — |
| — | Reserved | 84 | 0150h | — | × | × | × | — | — | — |
| — | Reserved | 85 | 0154h | — | × | × | × | — | — | — |
| — | Reserved | 86 | 0158h | — | × | × | × | — | — | — |
| — | Reserved | 87 | 015Ch | — | × | × | × | — | — | — |
| LVD | LVD1 | 88 | 0160h | Edge | ○ | × | ○ | IER0B.IEN0 | IPR088 | — |
| | LVD2 | 89 | 0164h | Edge | ○ | × | ○ | IER0B.IEN1 | IPR089 | — |
| USB0 | USBR0 | 90 | 0168h | Level | ○ | × | ○ | IER0B.IEN2 | IPR090 | — |
| — | Reserved | 91 | 016Ch | — | × | × | × | — | — | — |
| RTC | ALM | 92 | 0170h | Edge | ○ | × | ○ | IER0B.IEN4 | IPR092 | — |
| | PRD | 93 | 0174h | Edge | ○ | × | ○ | IER0B.IEN5 | IPR093 | — |
| — | Reserved | 94 | 0178h | — | × | × | × | — | — | — |
| — | Reserved | 95 | 017Ch | — | × | × | × | — | — | — |
| — | Reserved | 96 | 0180h | — | × | × | × | — | — | — |
| — | Reserved | 97 | 0184h | — | × | × | × | — | — | — |
| — | Reserved | 98 | 0188h | — | × | × | × | — | — | — |
| — | Reserved | 99 | 018Ch | — | × | × | × | — | — | — |
| — | Reserved | 100 | 0190h | — | × | × | × | — | — | — |
| — | Reserved | 101 | 0194h | — | × | × | × | — | — | — |
| S12AD | S12ADI0 | 102 | 0198h | Edge | ○ | ○ | × | IER0C.IEN6 | IPR102 | DTCER102 |
| | GBADI | 103 | 019Ch | Edge | ○ | ○ | × | IER0C.IEN7 | IPR103 | DTCER103 |
| — | Reserved | 104 | 01A0h | — | × | × | × | — | — | — |
| — | Reserved | 105 | 01A4h | — | × | × | × | — | — | — |
| ELC | ELSR18I | 106 | 01A8h | Edge | ○ | ○ | × | IER0D.IEN2 | IPR106 | DTCER106 |

**Table 14.3 Interrupt Vector Table (4/7)**

| Source of Interrupt Request Generation | Name | Vector No.*1 | Vector Address Offset | Form of Interrupt Detection | CPU | DTC | sstb Return | IER | IPR | DTCER |
|---|---|---|---|---|---|---|---|---|---|---|
| — | Reserved | 107 | 01ACh | — | × | × | × | — | — | — |
| — | Reserved | 108 | 01B0h | — | × | × | × | — | — | — |
| — | Reserved | 109 | 01B4h | — | × | × | × | — | — | — |
| — | Reserved | 110 | 01B8h | — | × | × | × | — | — | — |
| — | Reserved | 111 | 01BCh | — | × | × | × | — | — | — |
| — | Reserved | 112 | 01C0h | — | × | × | × | — | — | — |
| — | Reserved | 113 | 01C4h | — | × | × | × | — | — | — |
| MTU0 | TGIA0 | 114 | 01C8h | Edge | ○ | ○ | × | IER0E.IEN2 | IPR114 | DTCER114 |
| | TGIB0 | 115 | 01CCh | Edge | ○ | ○ | × | IER0E.IEN3 | | DTCER115 |
| | TGIC0 | 116 | 01D0h | Edge | ○ | ○ | × | IER0E.IEN4 | | DTCER116 |
| | TGID0 | 117 | 01D4h | Edge | ○ | ○ | × | IER0E.IEN5 | | DTCER117 |
| | TCIV0 | 118 | 01D8h | Edge | ○ | × | × | IER0E.IEN6 | IPR118 | — |
| | TGIE0 | 119 | 01DCh | Edge | ○ | × | × | IER0E.IEN7 | | — |
| | TGIF0 | 120 | 01E0h | Edge | ○ | × | × | IER0F.IEN0 | | — |
| MTU1 | TGIA1 | 121 | 01E4h | Edge | ○ | ○ | × | IER0F.IEN1 | IPR121 | DTCER121 |
| | TGIB1 | 122 | 01E8h | Edge | ○ | ○ | × | IER0F.IEN2 | | DTCER122 |
| | TCIV1 | 123 | 01ECh | Edge | ○ | × | × | IER0F.IEN3 | IPR123 | — |
| | TCIU1 | 124 | 01F0h | Edge | ○ | × | × | IER0F.IEN4 | | — |
| MTU2 | TGIA2 | 125 | 01F4h | Edge | ○ | ○ | × | IER0F.IEN5 | IPR125 | DTCER125 |
| | TGIB2 | 126 | 01F8h | Edge | ○ | ○ | × | IER0F.IEN6 | | DTCER126 |
| | TCIV2 | 127 | 01FCh | Edge | ○ | × | × | IER0F.IEN7 | IPR127 | — |
| | TCIU2 | 128 | 0200h | Edge | ○ | × | × | IER10.IEN0 | | — |
| — | Reserved | 129 | 0204h | — | × | × | × | — | — | — |
| — | Reserved | 130 | 0208h | — | × | × | × | — | — | — |
| — | Reserved | 131 | 020Ch | — | × | × | × | — | — | — |
| — | Reserved | 132 | 0210h | — | × | × | × | — | — | — |
| — | Reserved | 133 | 0214h | — | × | × | × | — | — | — |
| — | Reserved | 134 | 0218h | — | × | × | × | — | — | — |
| — | Reserved | 135 | 021Ch | — | × | × | × | — | — | — |
| — | Reserved | 136 | 0220h | — | × | × | × | — | — | — |
| — | Reserved | 137 | 0224h | — | × | × | × | — | — | — |
| — | Reserved | 138 | 0228h | — | × | × | × | — | — | — |
| MTU3 | TGIA3 | 129 | 0204h | Edge | ○ | ○ | × | IER10.IEN1 | IPR129 | DTCER129 |
| | TGIB3 | 130 | 0208h | Edge | ○ | ○ | × | IER10.IEN2 | | DTCER130 |
| | TGIC3 | 131 | 020Ch | Edge | ○ | ○ | × | IER10.IEN3 | | DTCER131 |
| | TGID3 | 132 | 0210h | Edge | ○ | ○ | × | IER10.IEN4 | | DTCER132 |
| | TCIV3 | 133 | 0214h | Edge | ○ | × | × | IER10.IEN5 | IPR133 | — |
| MTU4 | TGIA4 | 134 | 0218h | Edge | ○ | ○ | × | IER10.IEN6 | IPR134 | DTCER134 |
| | TGIB4 | 135 | 021Ch | Edge | ○ | ○ | × | IER10.IEN7 | | DTCER135 |
| | TGIC4 | 136 | 0220h | Edge | ○ | ○ | × | IER11.IEN0 | | DTCER136 |
| | TGID4 | 137 | 0224h | Edge | ○ | ○ | × | IER11.IEN1 | | DTCER137 |
| | TCIV4 | 138 | 0228h | Edge | ○ | ○ | × | IER11.IEN2 | IPR138 | DTCER138 |

**Table 14.3 Interrupt Vector Table (5/7)**

| Source of Interrupt Request Generation | Name | Vector No.*1 | Vector Address Offset | Form of Interrupt Detection | CPU | DTC | sstb Return | IER | IPR | DTCER |
|---|---|---|---|---|---|---|---|---|---|---|
| MTU5 | TGIU5 | 139 | 022Ch | Edge | ○ | ○ | × | IER11.IEN3 | IPR139 | DTCER139 |
| | TGIV5 | 140 | 0230h | Edge | ○ | ○ | × | IER11.IEN4 | | DTCER140 |
| | TGIW5 | 141 | 0234h | Edge | ○ | ○ | × | IER11.IEN5 | | DTCER141 |
| — | Reserved | 142 | 0238h | — | × | × | × | — | — | — |
| — | Reserved | 143 | 023Ch | — | × | × | × | — | — | — |
| — | Reserved | 144 | 0240h | — | × | × | × | — | — | — |
| — | Reserved | 145 | 0244h | — | × | × | × | — | — | — |
| — | Reserved | 146 | 0248h | — | × | × | × | — | — | — |
| — | Reserved | 147 | 024Ch | — | × | × | × | — | — | — |
| — | Reserved | 148 | 0250h | — | × | × | × | — | — | — |
| — | Reserved | 149 | 0254h | — | × | × | × | — | — | — |
| — | Reserved | 150 | 0258h | — | × | × | × | — | — | — |
| — | Reserved | 151 | 025Ch | — | × | × | × | — | — | — |
| — | Reserved | 152 | 0260h | — | × | × | × | — | — | — |
| — | Reserved | 153 | 0264h | — | × | × | × | — | — | — |
| — | Reserved | 154 | 0268h | — | × | × | × | — | — | — |
| — | Reserved | 155 | 026Ch | — | × | × | × | — | — | — |
| — | Reserved | 156 | 0270h | — | × | × | × | — | — | — |
| — | Reserved | 157 | 0274h | — | × | × | × | — | — | — |
| — | Reserved | 158 | 0278h | — | × | × | × | — | — | — |
| — | Reserved | 159 | 027Ch | — | × | × | × | — | — | — |
| — | Reserved | 160 | 0280h | — | × | × | × | — | — | — |
| — | Reserved | 161 | 0284h | — | × | × | × | — | — | — |
| — | Reserved | 162 | 0288h | — | × | × | × | — | — | — |
| — | Reserved | 163 | 028Ch | — | × | × | × | — | — | — |
| — | Reserved | 164 | 0290h | — | × | × | × | — | — | — |
| — | Reserved | 165 | 0294h | — | × | × | × | — | — | — |
| — | Reserved | 166 | 0298h | — | × | × | × | — | — | — |
| — | Reserved | 167 | 029Ch | — | × | × | × | — | — | — |
| — | Reserved | 168 | 02A0h | — | × | × | × | — | — | — |
| — | Reserved | 169 | 02A4h | — | × | × | × | — | — | — |
| POE | OEI1 | 170 | 02A8h | Level | ○ | × | × | IER15.IEN2 | IPR170 | — |
| | OEI2 | 171 | 02ACh | Level | ○ | × | × | IER15.IEN3 | IPR171 | — |
| — | Reserved | 172 | 02B0h | — | × | × | × | — | — | — |
| — | Reserved | 173 | 02B4h | — | × | × | × | — | — | — |
| — | Reserved | 174 | 02B8h | — | × | × | × | — | — | — |
| — | Reserved | 175 | 02BCh | — | × | × | × | — | — | — |
| — | Reserved | 176 | 02C0h | — | × | × | × | — | — | — |
| — | Reserved | 177 | 02C4h | — | × | × | × | — | — | — |
| — | Reserved | 178 | 02C8h | — | × | × | × | — | — | — |
| — | Reserved | 179 | 02CCh | — | × | × | × | — | — | — |
| — | Reserved | 180 | 02D0h | — | × | × | × | — | — | — |
| — | Reserved | 181 | 02D4h | — | × | × | × | — | — | — |

**Table 14.3 Interrupt Vector Table (6/7)**

| Source of Interrupt Request Generation | Name | Vector No.*1 | Vector Address Offset | Form of Interrupt Detection | CPU | DTC | sstb Return | IER | IPR | DTCER |
|---|---|---|---|---|---|---|---|---|---|---|
| — | Reserved | 182 | 02D8h | — | × | × | × | — | — | — |
| — | Reserved | 183 | 02DCh | — | × | × | × | — | — | — |
| — | Reserved | 184 | 02E0h | — | × | × | × | — | — | — |
| — | Reserved | 185 | 02E4h | — | × | × | × | — | — | — |
| — | Reserved | 186 | 02E8h | — | × | × | × | — | — | — |
| — | Reserved | 187 | 02ECh | — | × | × | × | — | — | — |
| — | Reserved | 188 | 02F0h | — | × | × | × | — | — | — |
| — | Reserved | 189 | 02F4h | — | × | × | × | — | — | — |
| — | Reserved | 190 | 02F8h | — | × | × | × | — | — | — |
| — | Reserved | 191 | 02FCh | — | × | × | × | — | — | — |
| — | Reserved | 192 | 0300h | — | × | × | × | — | — | — |
| — | Reserved | 193 | 0304h | — | × | × | × | — | — | — |
| — | Reserved | 194 | 0308h | — | × | × | × | — | — | — |
| — | Reserved | 195 | 030Ch | — | × | × | × | — | — | — |
| — | Reserved | 196 | 0310h | — | × | × | × | — | — | — |
| — | Reserved | 197 | 0314h | — | × | × | × | — | — | — |
| — | Reserved | 198 | 0318h | — | × | × | × | — | — | — |
| — | Reserved | 199 | 031Ch | — | × | × | × | — | — | — |
| — | Reserved | 200 | 0320h | — | × | × | × | — | — | — |
| — | Reserved | 201 | 0324h | — | × | × | × | — | — | — |
| — | Reserved | 202 | 0328h | — | × | × | × | — | — | — |
| — | Reserved | 203 | 032Ch | — | × | × | × | — | — | — |
| — | Reserved | 204 | 0330h | — | × | × | × | — | — | — |
| — | Reserved | 205 | 0334h | — | × | × | × | — | — | — |
| — | Reserved | 206 | 0338h | — | × | × | × | — | — | — |
| — | Reserved | 207 | 033Ch | — | × | × | × | — | — | — |
| — | Reserved | 208 | 0340h | — | × | × | × | — | — | — |
| — | Reserved | 209 | 0344h | — | × | × | × | — | — | — |
| — | Reserved | 210 | 0348h | — | × | × | × | — | — | — |
| — | Reserved | 211 | 034Ch | — | × | × | × | — | — | — |
| — | Reserved | 212 | 0350h | — | × | × | × | — | — | — |
| — | Reserved | 213 | 0354h | — | × | × | × | — | — | — |
| — | Reserved | 214 | 0358h | — | × | × | × | — | — | — |
| — | Reserved | 215 | 035Ch | — | × | × | × | — | — | — |
| — | Reserved | 216 | 0360h | — | × | × | × | — | — | — |
| — | Reserved | 217 | 0364h | — | × | × | × | — | — | — |
| SCI1 | ERI1 | 218 | 0368h | Level | ○ | × | × | IER1B.IEN2 | IPR218 | — |
| | RXI1 | 219 | 036Ch | Edge | ○ | ○ | × | IER1B.IEN3 | | DTCER219 |
| | TXI1 | 220 | 0370h | Edge | ○ | ○ | × | IER1B.IEN4 | | DTCER220 |
| | TEI1 | 221 | 0374h | Level | ○ | × | × | IER1B.IEN5 | | — |

**Table 14.3 Interrupt Vector Table (7/7)**

| Source of Interrupt Request Generation | Name | Vector No.*1 | Vector Address Offset | Form of Interrupt Detection | CPU | DTC | sstb Return | IER | IPR | DTCER |
|---|---|---|---|---|---|---|---|---|---|---|
| SCI5 | ERI5 | 222 | 0378h | Level | ○ | × | × | IER1B.IEN6 | IPR222 | — |
| | RXI5 | 223 | 037Ch | Edge | ○ | ○ | × | IER1B.IEN7 | | DTCER223 |
| | TXI5 | 224 | 0380h | Edge | ○ | ○ | × | IER1C.IEN0 | | DTCER224 |
| | TEI5 | 225 | 0384h | Level | ○ | × | × | IER1C.IEN1 | | — |
| — | Reserved | 226 | 0388h | — | × | × | × | — | — | — |
| — | Reserved | 227 | 038Ch | — | × | × | × | — | — | — |
| — | Reserved | 228 | 0390h | — | × | × | × | — | — | — |
| — | Reserved | 229 | 0394h | — | × | × | × | — | — | — |
| — | Reserved | 230 | 0398h | — | × | × | × | — | — | — |
| — | Reserved | 231 | 039Ch | — | × | × | × | — | — | — |
| — | Reserved | 232 | 03A0h | — | × | × | × | — | — | — |
| — | Reserved | 233 | 03A4h | — | × | × | × | — | — | — |
| — | Reserved | 234 | 03A8h | — | × | × | × | — | — | — |
| — | Reserved | 235 | 03ACh | — | × | × | × | — | — | — |
| — | Reserved | 236 | 03B0h | — | × | × | × | — | — | — |
| — | Reserved | 237 | 03B4h | — | × | × | × | — | — | — |
| SCI12 | ERI12 | 238 | 03B8h | Level | ○ | × | × | IER1D.IEN6 | IPR238 | — |
| | RXI12 | 239 | 03BCh | Edge | ○ | ○ | × | IER1D.IEN7 | | DTCER239 |
| | TXI12 | 240 | 03C0h | Edge | ○ | ○ | × | IER1E.IEN0 | | DTCER240 |
| | TEI12 | 241 | 03C4h | Level | ○ | × | × | IER1E.IEN1 | | — |
| | SCIX0 | 242 | 03C8h | Level | ○ | × | × | IER1E.IEN2 | IPR242 | — |
| | SCIX1 | 243 | 03CCh | Level | ○ | × | × | IER1E.IEN3 | IPR243 | — |
| | SCIX2 | 244 | 03D0h | Level | ○ | × | × | IER1E.IEN4 | IPR244 | — |
| | SCIX3 | 245 | 03D4h | Level | ○ | × | × | IER1E.IEN5 | IPR245 | — |
| RIIC0 | EEI0 | 246 | 03D8h | Level | ○ | × | × | IER1E.IEN6 | IPR246 | — |
| | RXI0 | 247 | 03DCh | Edge | ○ | ○ | × | IER1E.IEN7 | IPR247 | DTCER247 |
| | TXI0 | 248 | 03E0h | Edge | ○ | ○ | × | IER1F.IEN0 | IPR248 | DTCER248 |
| | TEI0 | 249 | 03E4h | Level | ○ | × | × | IER1F.IEN1 | IPR249 | — |
| — | Reserved | 250 | 03E8h | — | × | × | × | — | — | — |
| — | Reserved | 251 | 03ECh | — | × | × | × | — | — | — |
| — | Reserved | 252 | 03F0h | — | × | × | × | — | — | — |
| — | Reserved | 253 | 03F4h | — | × | × | × | — | — | — |
| — | Reserved | 254 | 03F8h | — | × | × | × | — | — | — |
| — | Reserved | 255 | 03FCh | — | × | × | × | — | — | — |

Note 1. An interrupt source with a smaller vector number takes precedence.

### 14.3.2 Fast Interrupt Vector Table

The address of the entry in the interrupt vector table that corresponds to the vector number of the fast interrupt is placed in the fast interrupt vector register (FINTV) of the CPU.

### 14.3.3 Non-maskable Interrupt Vector Table

The non-maskable interrupt vector table area is FFFF FFF8h.

## 14.4 Interrupt Operation

The interrupt controller performs the following processing.

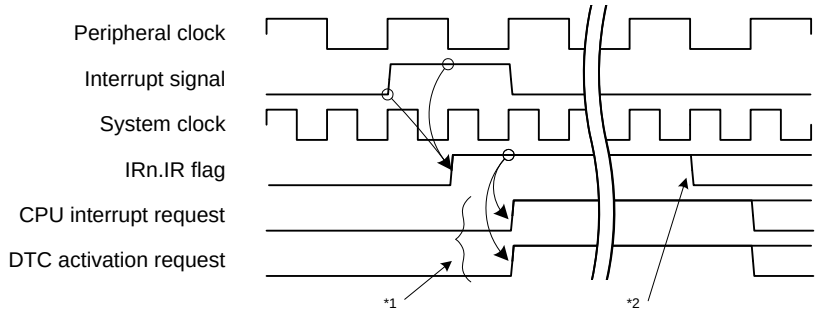
- Detecting interrupts
- Enabling and disabling interrupts
- Selecting interrupt request destinations (CPU interrupt, DTC activation)
- Determining priority

### 14.4.1 Detecting Interrupts

Interrupt requests are detected in either of two ways: the detection of edges of the interrupt signal or the detection of a level of the interrupt signal.
Edge detection or level detection is selected for the IRQi pins (i = 0 to 7) as external interrupt requests by the setting of the IRQCRi.IRQMD[1:0] bits.
For interrupts from peripheral modules, either edge detection or level detection is determined for each interrupt source.
For the correspondence between interrupt sources and methods of detection, see Table 14.3, Interrupt Vector Table.

#### 14.4.1.1 Operation of Status Flags for Edge-Detected Interrupts

Figure 14.2 shows the operation of the IRn.IR flag in the case of edge detection of an interrupt from a peripheral module or on an external pin.
The IRn.IR flag is set to 1 immediately after the transition of the interrupt signal due to generation of the interrupt. If the CPU is the request destination for the interrupt, the IR flag is automatically cleared to 0 on acceptance of the interrupt. If the DTC is the request destination for the interrupt, the IRn.IR flag operation differs according to the DTC transfer settings and transfer count. For details, see Table 14.4, Operation at DTC Activation. It is not necessary to clear the IRn.IR flag by software.




Note 1. One of the following requests is issued: CPU interrupt request, DTC activation request. For details of the setting, refer to section 14.4.3, Selecting Interrupt Request Destinations.
Note 2. When the CPU interrupt request is specified, this flag is set to 0 on acceptance of a CPU interrupt. For the timing of 0 setting at the DTC activation, see Table 14.4 Operation at DTC Activation.

**Figure 14.2 IRn.IR Flag Operation for Edge Detection Interrupts**

Figure 14.3 to Figure 14.5 show the interrupt signals of the interrupt controller. Note that the timings of the interrupts with interrupt vector numbers 64 to 95 are different from those of other interrupts. For the IRQ pin interrupts with interrupt vector numbers 64 to 79, “internal delay + 2 PCLK cycles” of delay is added after the IRQ pin input. For the interrupts with interrupt vector numbers 80 to 95, “2 PCLK cycles” of delay is added.

While the IRn.IR flag is 1 after an interrupt request is generated, the interrupt request that is generated again will be ignored.*1
Figure 14.3 shows the timing for IRn.IR flag re-setting.

Note 1. When the transmission or reception interrupt of the SCI, RSPI, or RIIC is generated with the IRn.IR flag being 1, the interrupt request is retained. After the IRn.IR flag is cleared to 0, the IRn.IR flag is set to 1 again by the retained request. For details, refer to descriptions of the interrupts in section 26, Serial Communications Interface (SCIe, SCIf), section 27, I²C Bus Interface (RIIC), and section 28, Serial Peripheral Interface (RSPI).

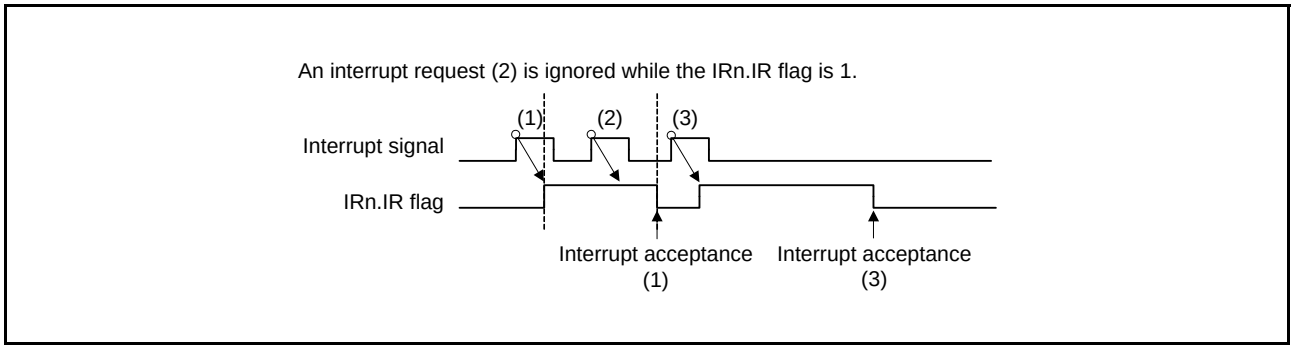



**Figure 14.3 Timing for IRn.IR Flag ReSetting**

If an interrupt is disabled after the IRn.IR flag is set to 1 (output of the interrupt request is disabled by the interrupt enable bit of the relevant peripheral module), the IRn.IR flag is not affected but retains its state. Figure 14.4 shows operation when the interrupt is disabled.

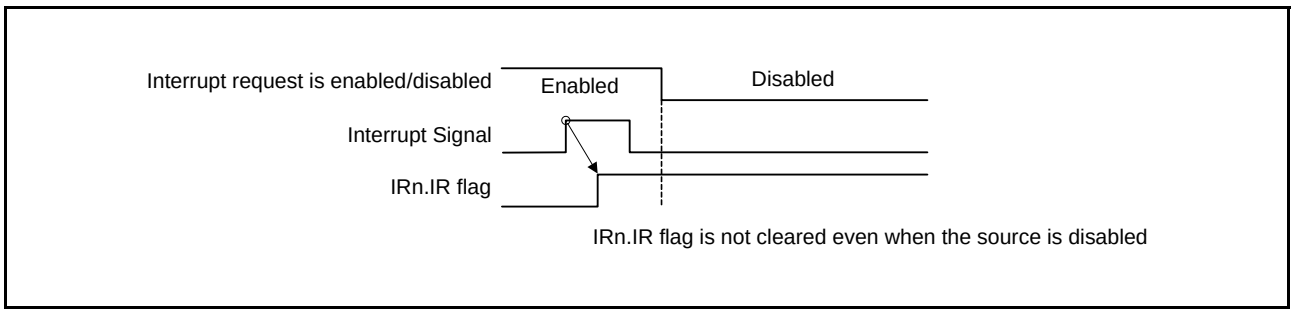



**Figure 14.4 Relationship between IRn.IR Flag Operation and Disabling of Interrupt Request**

### 14.4.1.2 Operation of Status Flags for Level-Detected Interrupts

Figure 14.5 shows the operation of the interrupt status flag (IR flag) in IRn in the case of level detection of an interrupt from a peripheral module or an external pin.

The IRn.IR flag remains set to 1 as long as the interrupt signal is asserted. To clear the IRn.IR flag to 0, clear the interrupt request in the source generating the interrupt. Confirm that the interrupt request flag in the source generating the interrupt has been cleared to 0 and that the IRn.IR flag has been cleared to 0, and then complete the interrupt handling.

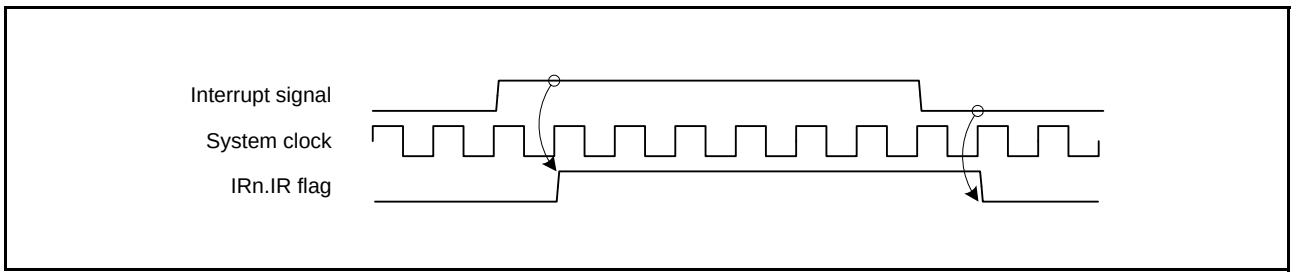



**Figure 14.5 IRn.IR Flag Operation for Level Detection Interrupts**

Figure 14.6 shows the procedure for handling level detection interrupts.

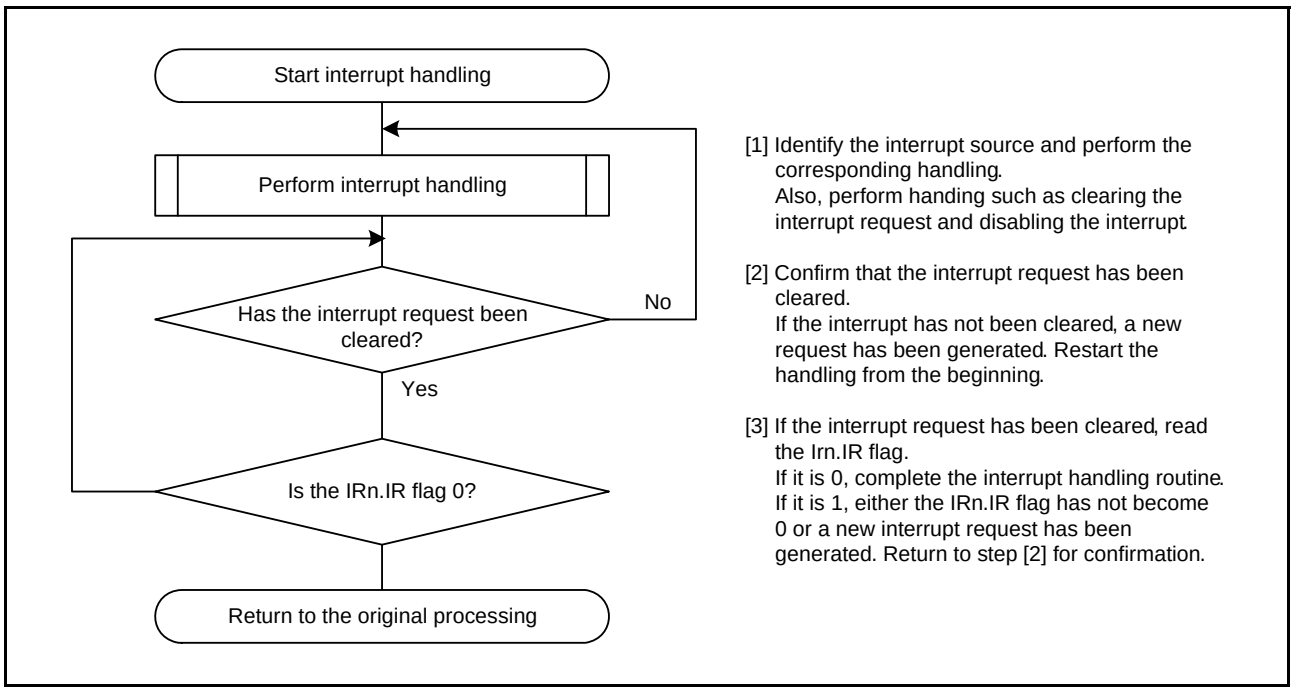



**Figure 14.6 Procedure for Handling Level Detection Interrupts**

### 14.4.2 Enabling and Disabling Interrupt Sources

Enabling requests from a given interrupt source requires the following settings.

1. In the case of interrupt requests from peripheral modules, setting the interrupt enable bit for the peripheral module to permit the output of interrupt requests from the source
2. Enabling of the interrupt by the IERm.IENj bit

When an interrupt request that is enabled at the corresponding source is generated, the corresponding IRn.IR flag is set to 1. Setting the IERm.IENj bit to enable an interrupt request allows the interrupt request for which the corresponding IRn.IR is 1 to be output to the interrupt request destination. Setting the IERm.IENj bit to disable an interrupt request suspends the output of the interrupt request for which the corresponding IRn.IR is 1.
The IRn.IR flag is not affected by the IERm.IENj bit.

Use the following procedure to disable interrupt requests.

1. Set the IERm.IENj bit to disable interrupt requests.
2. Set the peripheral module interrupt output enable bit to disable the output. Read the last written register and confirm that writing is completed.
3. Check the IRn.IR flag, and clear the IRn.IR flag if necessary.*1

Note 1. To disable the transmission or reception interrupt of the SCI, RSPI, or RIIC from the enabled state, clear the IRn.IR flag to 0 using the above procedure. For details, refer to descriptions of the interrupts in section 26, Serial Communications Interface (SCIe, SCIf), section 27, I2C Bus Interface (RIIC), and section 28, Serial Peripheral Interface (RSPI).

### 14.4.3 Selecting Interrupt Request Destinations

Possible settings for the request destination of each interrupt are fixed. That is, settings for request destination other than those listed in Table 14.3, Interrupt Vector Table, are not possible. Do not make an interrupt request destination setting that is not indicated by a O in Table 14.3.
If the DTC is selected as the destination for requests from an IRQ pin, be sure to set the IRQCRi.IRQMD[1:0] bits for that interrupt to select edge detection.
The following describes how to specify the destinations of interrupt requests.

#### (1) DTC Activation

Make the following settings for interrupt sources while the IERm.IENj bit is 0.

1. Set the DTC activation enable bit in the DTC activation enable register (DTCERn.DTCE) for the pertinent source to 1.

After making the above settings, set the IERm.IENj bit to 1.
In addition, set the DTC module start bit (DTCST.DTCST) to 1. The order of making settings for interrupt sources and enabling the DTC module start bit does not matter.
For the DTC setting procedure, refer to section 16.5, DTC Setting Procedure, in section 16, Data Transfer Controller (DTCa).

#### (2) CPU Interrupt Request

If the interrupt request destination is not the DTC, the interrupt request is sent to the CPU. Set the IERm.IENj bit to 1 while not the DTC activation settings described above are in place.

Table 14.4 shows operation when the DTC is the request destination.

**Table 14.4 Operation at DTC Activation**

| Interrupt Request Destination | DISEL | Remaining Number of Transfer Operations | Operation for Each Request | IR*1 | Interrupt Request Destination after Transfer |
|---|---|---|---|---|---|
| DTC*2 | 1 | ≠ 0 | DTC transfer → CPU interrupt | Cleared on interrupt acceptance by the CPU | DTC |
| | | = 0 | DTC transfer → CPU interrupt | Cleared on interrupt acceptance by the CPU | The DTCER.DTCE bit is cleared and the CPU becomes the destination. |
| | 0 | ≠ 0 | DTC transfer | Cleared at the start of DTC data transfer after reading DTC transfer information | DTC |
| | | = 0 | DTC transfer → CPU interrupt | Cleared on interrupt acceptance by the CPU | The DTCER.DTCE bit is cleared and the CPU becomes the destination. |

DISEL for the DTC is set by the DTC.MRB.DISEL bit.

Note 1. When the IRn.IR flag is 1, an interrupt request (DTC activation request) that is generated again will be ignored.

Note 2. For chain transfer, DTC transfer continues until the last chain transfer ends. Whether a CPU interrupt is generated at the end of chain transfer, the IRn.IR flag clear timing, and the interrupt request destination after transfer are determined by the state of DISEL and the remaining transfer count at the end of chain transfer. For the chain transfer, see Table 16.3, Chain Transfer Conditions in section 16, Data Transfer Controller (DTCa).

The request destination for an interrupt should be changed while the IERm.IENj bit is 0.

When a source is to be changed to an interrupt request or the DTC transfer information is to be changed while a transfer is not complete (i.e. while the DTCERn.DTCE bit has not been cleared) after the settings described under (1) DTC Activation have been made, follow the procedure below.

1. For both the source to be withdrawn and the source that will have a new target for activation, clear the IERm.IENj bits to 0.
2. Check the state of transfer by the DTC. If transfer is in progress, wait for its completion.
3. Make the settings described under (1) DTC Activation.

### 14.4.4 Determining Priority

Interrupt priority is determined for each interrupt request destination.
The priority for each interrupt request destination is determined as follows.

#### (1) Determining Priority When the CPU is the Request Destination of the Interrupt

A source selected for the fast interrupt has the highest priority. After that, an interrupt source with a larger value of the interrupt priority level select bits (IPRn.IPR[3:0]) takes priority. If interrupts with the same priority level are generated by multiple sources, the source with the smallest vector number takes precedence.

#### (2) Determining Priority When the DTC is the Request Destination of the Interrupt

The IPRn.IPR[3:0] bits have no effect. An interrupt source with a smaller vector number takes precedence.

### 14.4.5 Fast Interrupt

The fast interrupt is a facility for faster interrupt handling by the CPU, so is only effective for an interrupt request being conveyed to the CPU. That is, the fast interrupt setting has no effect on interrupt requests for the DTC.
The fast interrupt is set up by specifying the vector number of an interrupt source in the FIR.FVCT[7:0] bits and enabling the fast interrupt by setting the FIEN bit in FIR to 1. When the given source generates an interrupt, the interrupt is output to the CPU for handling as the fast interrupt.
The interrupt source selected for the fast interrupt has the highest priority regardless of the setting of the IPRn.IPR[3:0] bits.
For details on the fast interrupt, refer to section 13, Exception Handling.

### 14.4.6 Digital Filter

The digital filter function is provided for the external interrupt request IRQi pins (i = 0 to 7) and NMI pin interrupt.
The digital filter samples input signals at the filter sampling clock (PCLK) and removes the pulses of which length is less than three sampling cycles.
To use the digital filter for the IRQi pin, set the sampling clock cycle (PCLK, PCLK/8, PCLK/32, or PCLK/64) with the IRQFLTC0.FCLKSELi[1:0] bits (i = 0 to 7) and set the IRQFLTE0.FLTENi bit to 1 (digital filter enabled).
To use the digital filter for the NMI pin, set the sampling clock cycle (PCLK, PCLK/8, PCLK/32, or PCLK/64) with the NMIFLTC.NFCLKSEL[1:0] bits and set the NMIFLTE.NFLTEN bit to 1 (digital filter enabled).
Figure 14.7 shows an example of digital filter operation.

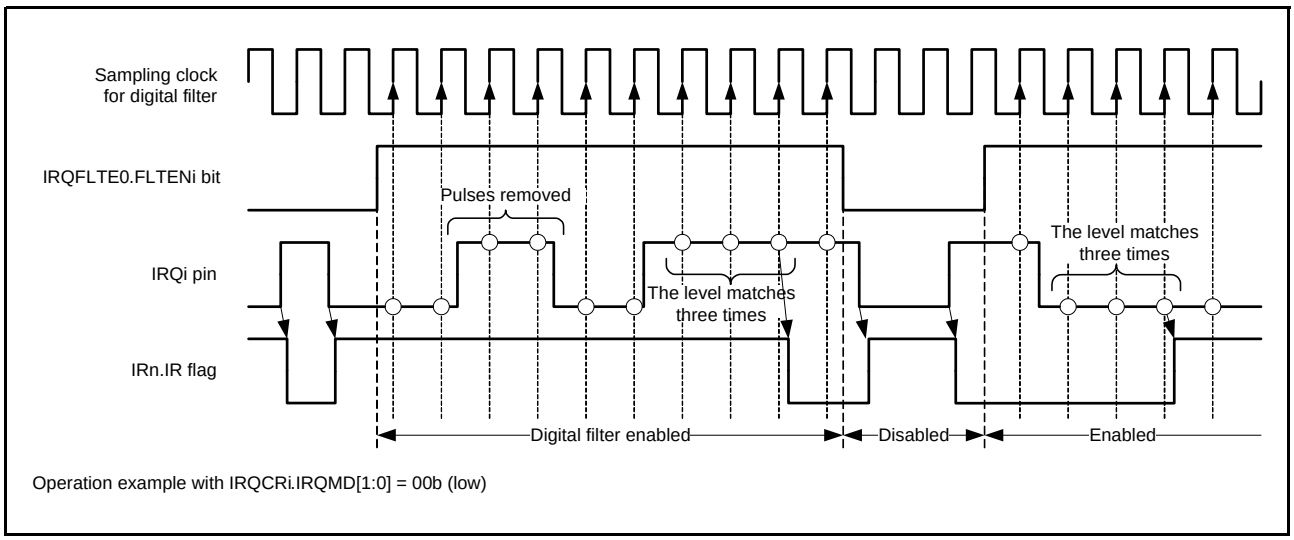



**Figure 14.7 Digital Filter Operation Example**

Before software standby mode is entered, set the IRQFLTE0.FLTENi and NMIFLTE.NFLTEN bits to 0 (digital filter disabled). To use the digital filter again after return from software standby mode, set the IRQFLTE0.FLTENi or NMIFLTE.NFLTEN bit to 1 (digital filter enabled).

### 14.4.7 External Pin Interrupts

The procedure for using the signal on an external pin as an interrupt is as follows.

1. Clear the IERm.IENj bit to 0 (interrupt request disabled).
2. Clear the IRQFLTE0.FLTENi bit (i = 0 to 7) to 0 (digital filter disabled).
3. Set the digital filter sampling clock with the IRQFLTC0.FCLKSELi[1:0] bits.
4. Make or confirm the I/O port settings.
5. Set the method of detection for the interrupt in the IRQCRi.IRQMD[1:0] bits.
6. Clear the corresponding IRn.IR flag to 0 (if edge detection is in use).
7. Set the IRQFLTE0.FLTENi bit to 1 (digital filter enabled).
8. If the interrupt is to be used for DTC activation, set the DTCERn.DTCE bit. (The interrupt will be a CPU interrupt if settings is not made.)
9. Set the IERm.IENj bit to 1 (interrupt request enabled).

## 14.5 Non-maskable Interrupt Operation

There are six types of non-maskable interrupt: the NMI pin interrupt, oscillation stop detection interrupt, IWDT underflow/refresh error, voltage monitoring 1 interrupt, and voltage monitoring 2 interrupt. Non-maskable interrupts are only usable as interrupts for the CPU; that is, they are not capable of DTC activation. Non-maskable interrupts take precedence over all interrupts, including the fast interrupt.
Non-maskable interrupt requests are accepted regardless of the states of the I (interrupt enable) bit and IPL[3:0] (processor interrupt priority level) bits in the PSW of the CPU. The current states of the non-maskable interrupts can be checked in the non-maskable interrupt status register (NMISR).
Confirm that all bits in the NMISR have returned to 0 in the non-maskable interrupt handler.
Non-maskable interrupts are disabled by default. If a system is to use non-maskable interrupts, the following procedure must be followed at the beginning of program processing.

Non-maskable interrupt usage procedure:

1. Set the stack pointer (SP).
2. To use the NMI pin, clear the NMIFLTE.NFLTEN bit to 0 (digital filter disabled).
3. To use the NMI pin, set the digital filter sampling clock with the NMIFLTC.NFCLKSEL[1:0] bits.
4. To use the NMI pin, set the NMI pin detection sense with the NMICR.NMIMD bit.
5. To use the NMI pin, write 1 to the NMICLR.NMICLR bit to clear the NMISR.NMIST flag to 0.
6. To use the NMI pin, set the NMIFLTE.NFLTEN bit to 1 (digital filter enabled).
7. Enable the non-maskable interrupt by writing 1 to the corresponding bit in the non-maskable interrupt enable register (NMIER).

After 1 is written to the NMIER register, subsequent write access to the NMIEN bit in NMIER is ignored. The NMI interrupt cannot be disabled. It can be disabled only by a reset.
For the flow of non-maskable interrupt handling, refer to section 13, Exception Handling.
Writing 1 to the NMICLR.NMICLR bit clears the NMI status flag (NMISR.NMIST) to 0.
Writing 1 to the NMICLR.OSTCLR bit clears the oscillation stop detection interrupt status flag (NMISR.OSTST) to 0.
Writing 1 to the NMICLR.IWDTCLR bit clears the IWDT underflow/refresh error status flag (NMISR.IWDTST) to 0.
Writing 1 to the NMICLR.LVD1CLR bit clears the voltage monitoring 1 interrupt status flag (NMISR.LVD1ST) to 0.
Writing 1 to the NMICLR.LVD2CLR bit clears the voltage monitoring 2 interrupt status flag (NMISR.LVD2ST) to 0.

## 14.6 Return from Power-Down States

The interrupt sources that can be used to return operation from sleep mode, deep sleep mode, or software standby mode are listed in Table 14.3, Interrupt Vector Table.
For details, refer to section 11, Low Power Consumption. The following describes how to use an interrupt to return operation from each low power consumption mode.

### 14.6.1 Return from Sleep Mode or Deep Sleep Mode

If the interrupt controller is to return operation from sleep mode in response to an interrupt or non-maskable interrupt, make the following settings for the interrupt.

- Interrupts

1. Select the CPU as the interrupt request destination.
2. Use the IERm.IENj bit to enable the given interrupt request.
3. Set a priority level higher than that set in the CPU.PSW.IPL[3:0] bits.

- Non-maskable interrupts

Use the NMIER register to enable the given interrupt request.

### 14.6.2 Return from Software Standby Mode

The interrupt controller can return operation from a non-maskable interrupt or an interrupt that enables the return from the software standby mode.
The conditions for the return are listed below.

- Interrupts

1. Select the interrupt source that enables the return from the software standby mode.
2. Select the CPU as the interrupt request destination.
3. Use the IERm.IENj bit to enable the given interrupt request.
4. Set a priority level higher than that set in the CPU.PSW.IPL[3:0] bits.
   (For the interrupt source specified as a fast interrupt, as well as setting the fast interrupt set register (FIR), the interrupt priority level (IPRn) should be set above the level set by CPU.PSW.IPL.)

Interrupt requests through the IRQ pins that do not satisfy the above conditions are not detected while the clock is stopped in software standby mode.

- Non-maskable interrupts

Use the NMIER register to enable the given interrupt request.

- Procedure to make a transition to/from software standby mode

1. Before software standby mode is entered, disable the digital filter for the interrupt source as a return target (IRQFLTE0.FLTENi = 0, NMIFLTE.NFLTEN = 0).
2. To use the digital filter again after return from software standby mode, enable the digital filter (IRQFLTE0.FLTENi = 1, NMIFLTE.NFLTEN = 1).

## 14.7 Usage Note

### 14.7.1 Note on WAIT Instruction Used with Non-Maskable Interrupt

Before executing the WAIT instruction, check to see that all the status flags in NMISR are 0.

# 15. Buses

## 15.1 Overview

Table 15.1 lists the bus specifications, Figure 15.1 shows the bus configuration, and Table 15.2 lists the addresses assigned to each bus.

**Table 15.1 Bus Specifications**

| Bus Type | | Description |
|---|---|---|
| CPU bus | Instruction bus | • Connected to the CPU for instructions<br>• Connected to on-chip memory (RAM, ROM)<br>• Operates in synchronization with the system clock (ICLK) |
| | Operand bus | • Connected to the CPU (for operands)<br>• Connected to on-chip memory (RAM, ROM)<br>• Operates in synchronization with the system clock (ICLK) |
| Memory bus | Memory bus 1 | • Connected to RAM |
| | Memory bus 2 | • Connected to ROM |
| Internal main buses | Internal main bus 1 | • Connected to the CPU<br>• Operates in synchronization with the system clock (ICLK) |
| | Internal main bus 2 | • Connected to the DTC<br>• Connected to on-chip memory (RAM, ROM)<br>• Operates in synchronization with the system clock (ICLK) |
| Internal peripheral buses | Internal peripheral bus 1 | • Connected to peripheral modules (DTC, interrupt controller, and bus error monitoring section)<br>• Operates in synchronization with the system clock (ICLK) |
| | Internal peripheral bus 2 | • Connected to peripheral modules<br>• Operates in synchronization with the peripheral module clock (PCLKB) |
| | Internal peripheral bus 3 | • Connected to peripheral modules (USB)<br>• Operates in synchronization with the peripheral module clock (PCLKB) |
| | Internal peripheral bus 6 | • Connected to ROM (P/E) and E2 DataFlash<br>• Operates in synchronization with the FlashIF clock (FCLK) |

P/E: Programming/Erasure

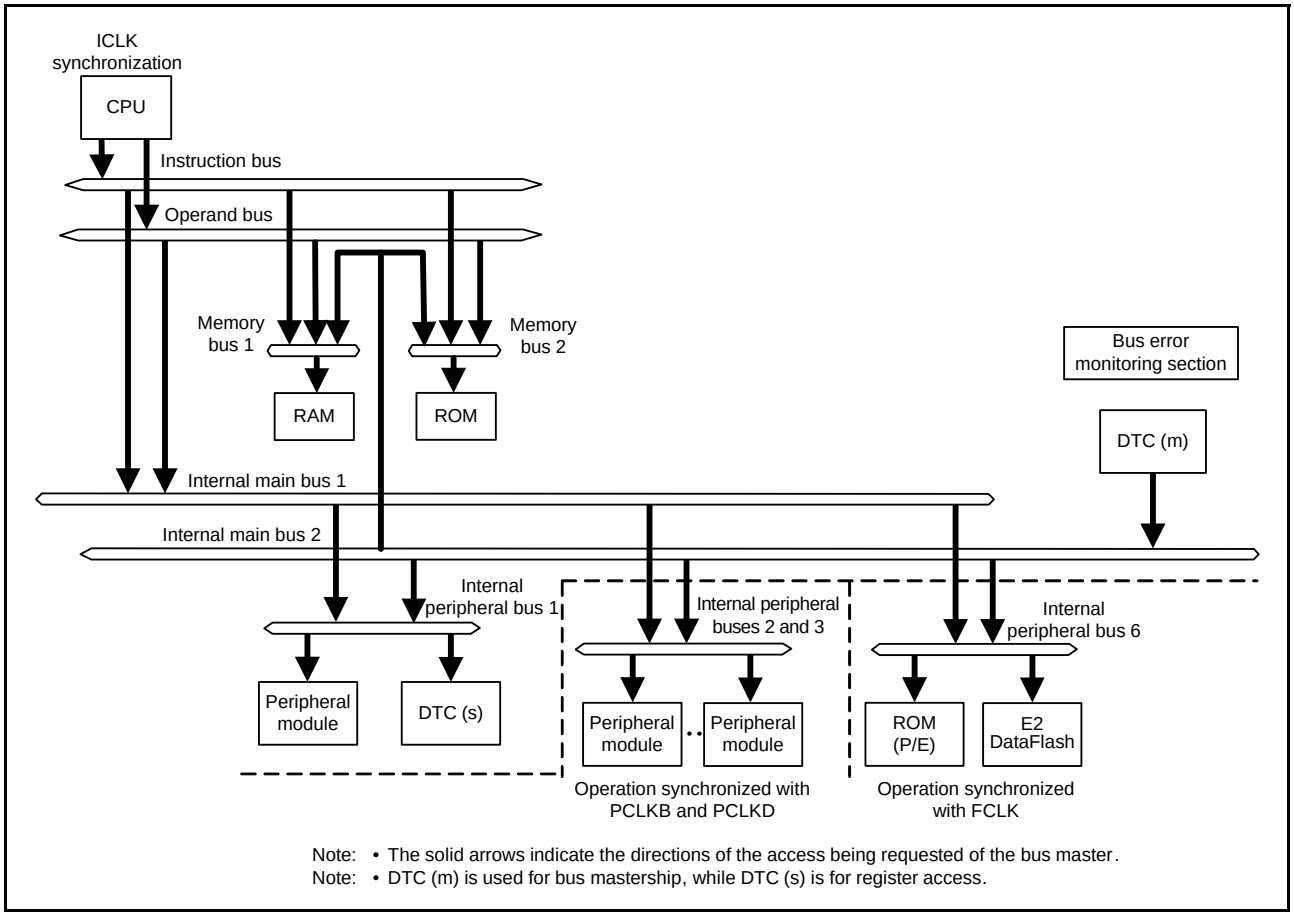



**Figure 15.1 Bus Configuration**

**Table 15.2 Addresses Assigned for Each Bus**

| Address | Bus | Area |
|---|---|---|
| 0000 0000h to 0000 FFFFh | Memory bus 1 | RAM |
| 0008 0000h to 0008 7FFFh | Internal peripheral bus 1 | Peripheral I/O registers |
| 0008 8000h to 0009 FFFFh | Internal peripheral bus 2 | |
| 000A 0000h to 000B FFFFh | Internal peripheral bus 3 | |
| 0010 0000h to 00FF FFFFh | Internal peripheral bus 6 | E2 DataFlash memory and ROM (for programming/erasure) |
| 8000 0000h to FEFF FFFFh | Memory bus 2 | ROM (for reading only) |
| FF00 0000h to FFFF FFFFh | | |

## 15.2 Description of Buses

### 15.2.1 CPU Buses

The CPU buses consist of the instruction and operand buses, which are connected to internal main bus 1. As the names suggest, the instruction bus is used to fetch instructions for the CPU, while the operand bus is used for operand access. Connection of the instruction and operand buses to RAM and ROM provides the CPU with direct access to these areas, i.e. access is not via internal main bus 1. However, only reading is possible in direct access to ROM by the CPU; programming and erasure are handled via an internal peripheral bus.
Bus requests for instruction fetching and operand access are arbitrated through internal main bus 1. The order of priority is operand access then instruction fetching.
If instruction fetching and operand access are requested for different buses (memory bus 1, memory bus 2, and internal main bus 1), the bus-access operations can proceed simultaneously. For example, parallel access to ROM and RAM is possible.

### 15.2.2 Memory Buses

The memory buses consist of memory bus 1 and memory bus 2. RAM is connected to memory bus 1 and ROM is connected to memory bus 2. Requests for bus mastership from the CPU buses (instruction fetching and operand) and internal main bus 2 are arbitrated through memory buses 1 and 2.
The priority order of CPU bus and internal main bus 2 can be set using the memory bus 1 (RAM) priority control bits (BPRA[1:0]) and memory bus 2 (ROM) priority control bits (BPRO[1:0]) in the bus priority control register (BUSPRI) for the corresponding memory buses. When the priority order is fixed, internal main bus 2 has priority over the CPU bus (operand over instruction fetching). When the priority order is toggled, the bus for which a request has been accepted has lower priority.

### 15.2.3 Internal Main Buses

The internal main buses consist of a bus for use by the CPU (internal main bus 1) and a bus for use by the other bus-master modules, i.e. the DTC (internal main bus 2).
Bus requests for instruction fetching and operand access are arbitrated through internal main bus 1. The order of priority is operand access then instruction fetching.
Requests for bus mastership from the DTC is arbitrated by internal main bus 2. The order of priority is as shown in Table 15.3.
If the CPU and another bus master are requesting access to different buses (on-chip memory, internal peripheral buses 1 to 3 and 6), the respective bus-access operations can proceed simultaneously.
However, when the CPU executes the XCHG instruction, requests for bus access from masters other than the CPU are not accepted until data transfer for the XCHG instruction is completed regardless of the bus priority control register (BUSPRI) setting. Furthermore, requests for bus access from masters other than the DTC are not accepted during reading and writing-back of transfer control information for the DTC.

**Table 15.3 Order of Priority for Bus Masters**

| Priority | Bus Master |
|---|---|
| High ↑ Low | DTC |
| | CPU |

### 15.2.4 Internal Peripheral Buses

Connection of peripheral modules to the internal peripheral buses is as described in Table 15.4.

**Table 15.4 Connection of Peripheral Modules to the Internal Peripheral Buses**

| Type of Bus | Peripheral Modules |
|---|---|
| Internal peripheral bus 1 | DTC, interrupt controller, and bus error monitoring section |
| Internal peripheral bus 2 | Peripheral modules other than those connected to internal peripheral buses 1 and 3 |
| Internal peripheral bus 3 | USB |
| Internal peripheral bus 6 | ROM (P/E)/E2 DataFlash |

Requests for bus mastership from the CPU (internal main bus 1) and other bus masters (internal main bus 2) are arbitrated through internal peripheral buses 1 to 3 and 6.
The priority order of the two internal main buses can be set using the bus priority control register (BUSPRI). The priority order can be set with the internal peripheral bus 1 priority control bits (BUSPRI.BPIB[1:0]), internal peripheral buses 2 and 3 priority control bits (BUSPRI.BPGB[1:0]), and internal peripheral bus 6 priority control bits (BUSPRI.BPFB[1:0]) for the corresponding internal peripheral buses. When the priority order is fixed, internal main bus 2 has priority over internal main bus 1. When the priority order is toggled, a bus has a lower priority when the request of that bus is accepted.
The order of accepting requests may change depending on the BUSPRI setting (see Figure 15.2).

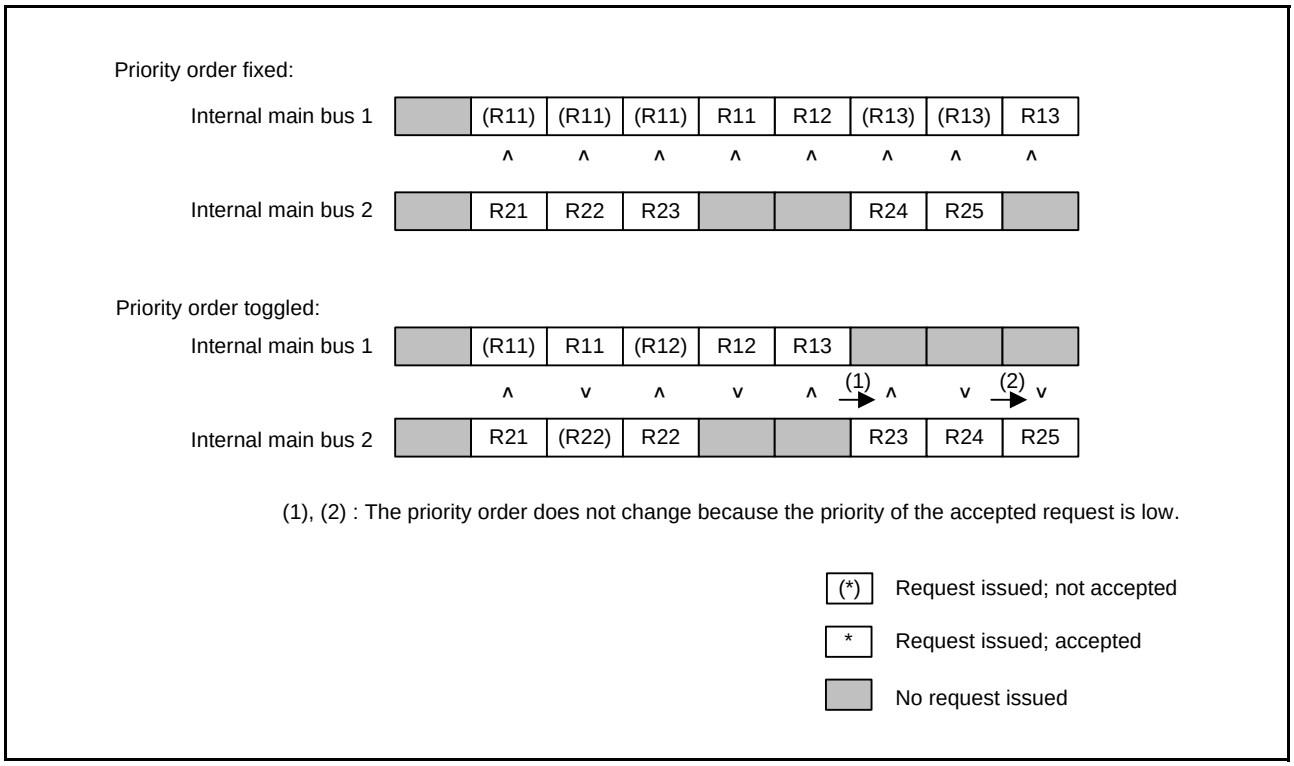



**Figure 15.2 Priority Order Between Internal Peripheral Bus Accesses**

### 15.2.5 Write Buffer Function (Internal Peripheral Bus)

The internal peripheral bus has the write buffer function, which allows the next round of bus access to start, before the current write access is completed, in write access. However, if the following round of bus access is from the same bus master but to the different internal peripheral bus, it is suspended until the bus operations already in progress are completed. When read access to the internal memory is scheduled after the write access to the internal peripheral bus from the CPU, the following round of bus access can be started before the current bus operation is completed and thus the order of accesses may be changed (see Figure 15.3).

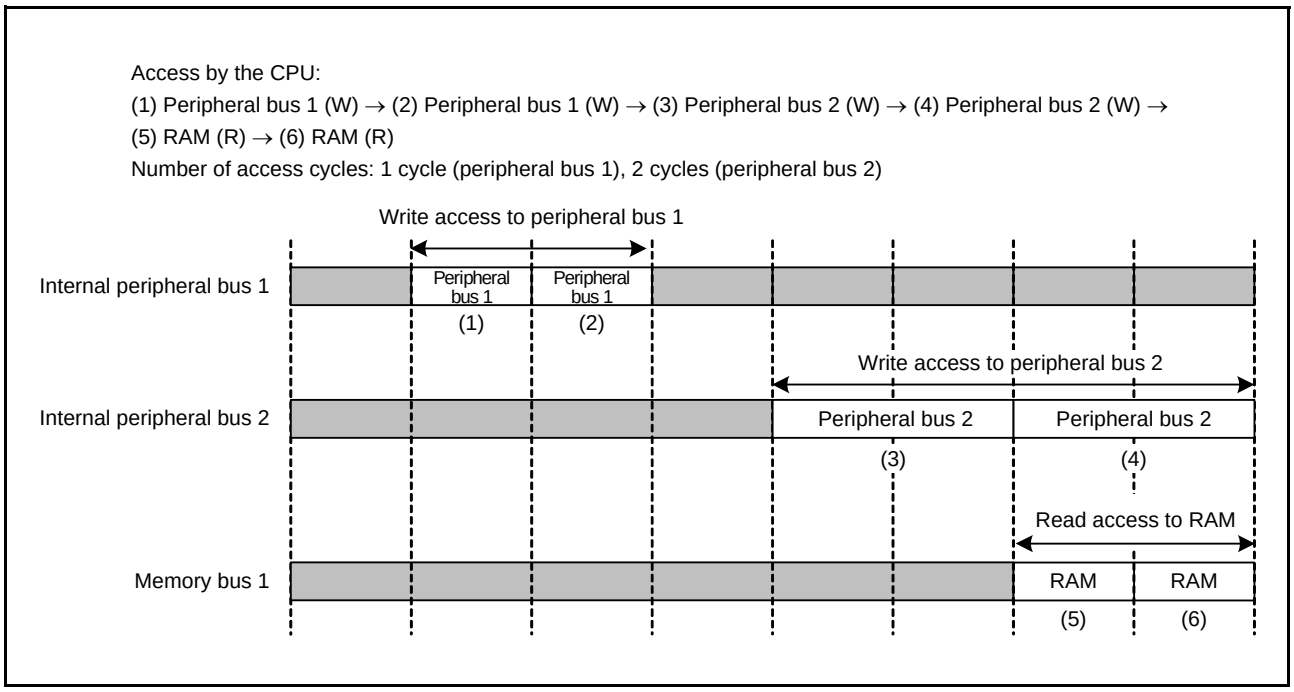



**Figure 15.3 Write Buffer Function**

### 15.2.6 Parallel Operation

Parallel operation is possible when different bus-master modules are requesting access to different slave modules. For example, if the CPU is fetching an instruction from ROM and an operand from RAM, the DTC is able to handle transfer between a peripheral bus and peripheral bus at the same time.
An example of parallel operations is shown in Figure 15.4. In this example, the CPU is able to employ the instruction and operand buses for simultaneous access to ROM and RAM, respectively. Furthermore, the DTC simultaneously employs internal main bus 2 for access to a peripheral bus during access to RAM and ROM by the CPU.

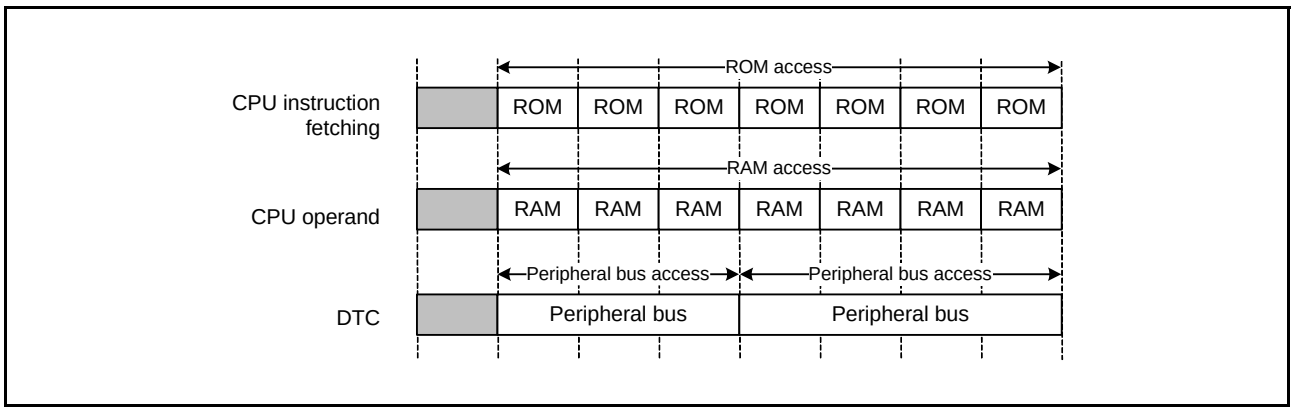



**Figure 15.4 Example of Parallel Operations**

### 15.2.7 Restrictions

#### (1) Prohibition of Access that Spans Multiple Areas of Address Space

Single access that spans two areas of the address space is prohibited, and operation of such an access is not guaranteed. Ensure that a single word or longword access does not span across two areas by crossing address space area boundaries.

#### (2) Restrictions on RMPA and String-Manipulation Instructions

(a) The allocation of data to be handled by RMPA or string-manipulation instructions to I/O registers is prohibited, and operation is not guaranteed if this restriction is not observed.

## 15.3 Register Descriptions

### 15.3.1 Bus Error Status Clear Register (BERCLR)

Address(es): 0008 1300h

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | STSCLR |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b0 | STSCLR | Status Clear | 0: Invalid<br>1: Bus error status register cleared | (W)*1 |
| b7 to b1 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |

Note 1. Only writing 1 is effective; i.e. writing 0 has no effect.

**STSCLR Bit (Status Clear)**

Writing 1 to this bit clears the bus error status registers 1 and 2 (BERSR1 and BERSR2).
Writing 0 has no effect. It is read as 0.

### 15.3.2 Bus Error Monitoring Enable Register (BEREN)

Address(es): 0008 1304h

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | IGAEN |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b0 | IGAEN | Illegal Address Access Detection Enable | 0: Illegal address access detection is disabled.<br>1: Illegal address access detection is enabled. | R/W |
| b7 to b1 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |

## 15.3.3 Bus Error Status Register 1 (BERSR1)

Address(es): 0008 1308h

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | MST[2:0] | | | — | — | — | IA |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b0 | IA | Illegal Address Access | 0: Illegal address access not made<br>1: Illegal address access made | R |
| b3 to b1 | — | Reserved | These bits are read as 0. Writing to these bits has no effect. | R |
| b6 to b4 | MST[2:0] | Bus Master Code | b6 b4<br>0 0 0: CPU<br>0 0 1: Reserved<br>0 1 0: Reserved<br>0 1 1: DTC<br>1 0 0: Reserved<br>1 0 1: Reserved<br>1 1 0: Reserved<br>1 1 1: Reserved | R |
| b7 | — | Reserved | This bit is read as 0. Writing to this bit has no effect. | R |

**MST[2:0] Bits (Bus Master Code)**

These bits indicate the bus master that accessed a bus when a bus error occurred.

## 15.3.4 Bus Error Status Register 2 (BERSR2)

Address(es): 0008 130Ah

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | ADDR[12:0] | | | | | | | | | | | | | — | — | — |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b2 to b0 | — | Reserved | These bits are read as 0. Writing to these bits has no effect. | R |
| b15 to b3 | ADDR[12:0] | Bus Error Occurrence Address | The upper 13 bits of an address that was accessed when a bus error occurred (in units of 512 Kbytes). | R |

## 15.3.5 Bus Priority Control Register (BUSPRI)

Address(es): 0008 1310h

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | BPFB[1:0] | | — | — | BPGB[1:0] | | BPIB[1:0] | | BPRO[1:0] | | BPRA[1:0] | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b1, b0 | BPRA[1:0] | Memory Bus 1 (RAM) Priority Control | b1 b0<br>0 0: The order of priority is fixed.<br>0 1: The order of priority is toggled.<br>1 0: Setting prohibited<br>1 1: Setting prohibited | R/(W) *1 |
| b3, b2 | BPRO[1:0] | Memory Bus 2 (ROM) Priority Control | b3 b2<br>0 0: The order of priority is fixed.<br>0 1: The order of priority is toggled.<br>1 0: Setting prohibited<br>1 1: Setting prohibited | R/(W) *1 |
| b5, b4 | BPIB[1:0] | Internal Peripheral Bus 1 Priority Control | b5 b4<br>0 0: The order of priority is fixed.<br>0 1: The order of priority is toggled.<br>1 0: Setting prohibited<br>1 1: Setting prohibited | R/(W) *1 |
| b7, b6 | BPGB[1:0] | Internal Peripheral Buses 2 and 3 Priority Control | b7 b6<br>0 0: The order of priority is fixed.<br>0 1: The order of priority is toggled.<br>1 0: Setting prohibited<br>1 1: Setting prohibited | R/(W) *1 |
| b9, b8 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |
| b11, b10 | BPFB[1:0] | Internal Peripheral Bus 6 Priority Control | b11 b10<br>0 0: The order of priority is fixed.<br>0 1: The order of priority is toggled.<br>1 0: Setting prohibited<br>1 1: Setting prohibited | R/(W) *1 |
| b15 to b12 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |

Note 1. These bits can be written to only once while the DTC is stopped. When they are written to more than one time, the operation is not guaranteed.

### BPRA[1:0] Bits (Memory Bus 1 (RAM) Priority Control)

These bits specify the priority order for memory bus 1 (RAM).

When the priority order is fixed, internal main bus 2 has priority over the CPU bus.

When the priority order is toggled, a bus has a lower priority when the request of that bus is accepted.

### BPRO[1:0] Bits (Memory Bus 2 (ROM) Priority Control)

These bits specify the priority order for memory bus 2 (ROM).

When the priority order is fixed, internal main bus 2 has priority over the CPU bus.

When the priority order is toggled, a bus has a lower priority when the request of that bus is accepted.

### BPIB[1:0] Bits (Internal Peripheral Bus 1 Priority Control)

These bits specify the priority order for internal peripheral bus 1.

When the priority order is fixed, internal main bus 2 has priority over internal main bus 1.

When the priority order is toggled, a bus has a lower priority when the request of that bus is accepted.

**BPGB[1:0] Bits (Internal Peripheral Buses 2 and 3 Priority Control)**

These bits specify the priority order for internal peripheral buses 2 and 3.
When the priority order is fixed, internal main bus 2 has priority over internal main bus 1.
When the priority order is toggled, a bus has a lower priority when the request of that bus is accepted.

**BPFB[1:0] Bits (Internal Peripheral Bus 6 Priority Control)**

These bits specify the priority order for internal peripheral bus 6.
When the priority order is fixed, internal main bus 2 has priority over internal main bus 1.
When the priority order is toggled, a bus has a lower priority when the request of that bus is accepted.

## 15.4 Bus Error Monitoring Section

The bus error monitoring section monitors the individual areas for bus errors, and when a bus error occurs, the error is indicated to the bus master.

### 15.4.1 Type of Bus Error

There is a illegal address access bus error.
Illegal address access is the detection of access to an illegal area.

#### 15.4.1.1 Illegal Address Access

When the illegal address access detection enable bit (IGAEN) in the bus error monitoring enable register (BEREN) is set to 1, access to an illegal address area leads to illegal address access errors.
The address ranges where access will lead to illegal address access errors are listed in Table 15.5.

### 15.4.2 Operations When a Bus Error Occurs

When a bus error occurs, the error is indicated to the CPU. Operation is not guaranteed when a bus error occurs.

- Bus error indication to the CPU
  An interrupt is generated. The ICU.IERn register can specify whether to generate an interrupt in the case of a bus error.

### 15.4.3 Conditions Leading to Bus Errors

Table 15.5 lists the type of bus errors for each area in the respective address space.
If an illegal address access error is detected when no bus error has occurred (bus error status register n (BERSRn; n = 1, 2) is cleared), the detected error is reflected in the BERSRn register. Once a bus error occurs, no subsequent bus errors are reflected in the register unless the register is cleared.
If bus errors are simultaneously caused by two or more bus masters, error information of only one bus master is reflected. Once a bus error occurs, the status is retained until the BERSRn register is cleared.

**Table 15.5 Type of Bus Errors**

| Address | Type of Area | Type of Error |
|---|---|---|
| | | Illegal Address Access |
| 0000 0000h to 0007 FFFFh | Memory bus 1 | — |
| 0008 0000h to 0008 7FFFh | Internal peripheral bus 1 | — |
| 0008 8000h to 0009 FFFFh | Internal peripheral bus 2 | Δ |
| 000A 0000h to 000B FFFFh | Internal peripheral bus 3 | Δ |
| 000C 0000h to 000E FFFFh | Reserved area | ○ |
| 0010 0000h to 00FF FFFFh | Internal peripheral bus 6 | Δ |
| 0100 0000h to 07FF FFFFh | Reserved area | ○ |
| 0800 0000h to 0FFF FFFFh | Reserved area | — |
| 1000 0000h to 7FFF FFFFh | Reserved area | ○ |
| 8000 0000h to FFFF FFFFh | Memory bus 2 | — |

—: A bus error does not result.
Δ: A bus error may or may not result.
○: A bus error results.
Note: • The capacity of the RAM, data flash, and ROM differs depending on the product. For details, refer to section 34, RAM, and section 35, Flash Memory.

# 16. Data Transfer Controller (DTCa)

This MCU incorporates a data transfer controller (DTC).
The DTC is activated by an interrupt request to perform data transfers.

## 16.1 Overview

Table 16.1 lists the specifications of the DTC, and Figure 16.1 shows a block diagram of the DTC.

**Table 16.1 DTC Specifications**

| Item | Description |
|---|---|
| Transfer modes | • Normal transfer mode<br>A single activation leads to a single data transfer.<br>• Repeat transfer mode<br>A single activation leads to a single data transfer.<br>The transfer address is returned to the transfer start address after the number of data transfers corresponding to “repeat size”.<br>The maximum number of repeat transfers is 256, and the maximum data transfer size is 256 × 32 bits, 1024 bytes.<br>• Block transfer mode<br>A single activation leads to the transfer of a single block.<br>The maximum block size is 256 × 32 bits = 1024 bytes. |
| Transfer channel | • Channel transfer corresponding to the interrupt source is possible (transferred by DTC activation request from the ICU).<br>• Data of multiple channels can be transferred on a single activation source (chain transfer).<br>• Either “executed when the counter is 0” or “always executed” can be selected for chain transfer. |
| Transfer space | • In short-address mode: 16 Mbytes<br>(Areas from 0000 0000h to 007F FFFFh and FF80 0000h to FFFF FFFFh except reserved areas)<br>• In full-address mode: 4 Gbytes<br>(Area from 0000 0000h to FFFF FFFFh except reserved areas) |
| Data transfer units | • Single data: 1 byte (8 bits), 1 word (16 bits), 1 longword (32 bits)<br>• Single block size: 1 to 256 data |
| CPU interrupt source | • An interrupt request can be generated to the CPU on a DTC activation interrupt.<br>• An interrupt request can be generated to the CPU after a single data transfer.<br>• An interrupt request can be generated to the CPU after data transfer of specified volume. |
| Event link function | Event link request is generated after one data transfer (for block, after one block transfer). |
| Read skip | Transfer information read skip can be specified. |
| Write-back skip | When “fixed” is selected for transfer source address or transfer destination address, write-back skip execution is provided. |
| Low power consumption function | Module stop state can be set. |

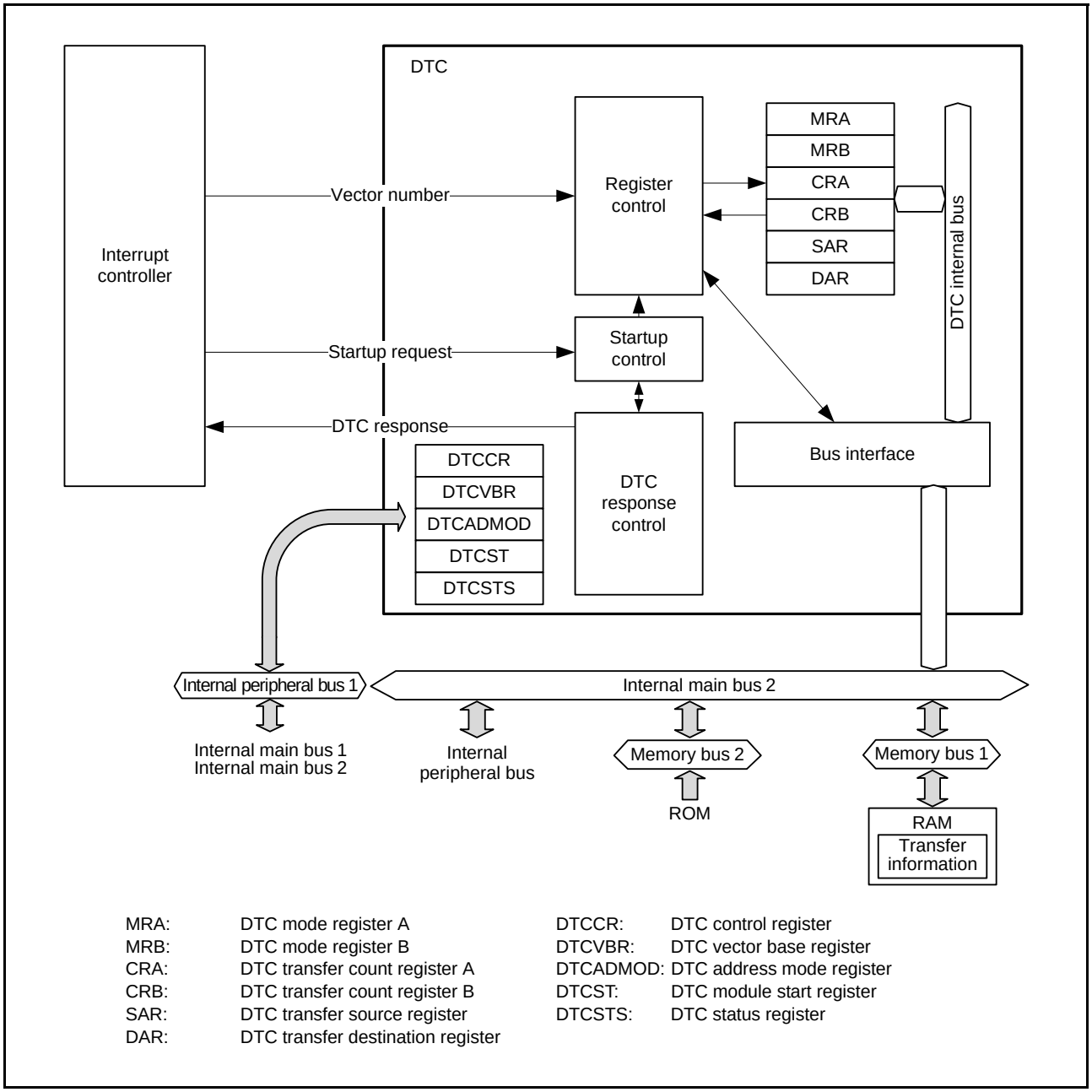



**Figure 16.1 Block Diagram of DTC**

## 16.2 Register Descriptions

Registers MRA, MRB, SAR, DAR, CRA, and CRB are DTC internal registers, which cannot be directly accessed from the CPU. Values to be set in the DTC internal registers are placed in the RAM area as transfer information. When an activation request is generated, the DTC reads the transfer information from the RAM area and set them in the internal registers. After the data transfer ends, the internal register contents are written back to the RAM area as transfer information.

### 16.2.1 DTC Mode Register A (MRA)

Address(es): (inaccessible directly from the CPU)

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | MD[1:0] | | SZ[1:0] | | SM[1:0] | | — | — |
| Value after reset: | x | x | x | x | x | x | x | x |

x: Undefined

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b1, b0 | — | Reserved | These bits are read as undefined. The write value should be 0. | — |
| b3, b2 | SM[1:0] | Transfer Source Address Addressing Mode | b3 b2<br>0 0: Address in the SAR register is fixed. (write-back to SAR is skipped.)<br>0 1: Address in the SAR register is fixed. (write-back to SAR is skipped.)<br>1 0: SAR value is incremented after data transfer. (+1 when SZ[1:0] bits = 00b, +2 when SZ[1:0] bits = 01b, +4 when SZ[1:0] bits = 10b)<br>1 1: SAR value is decremented after data transfer. (–1 when SZ[1:0] bits = 00b, –2 when SZ[1:0] bits = 01b, –4 when SZ[1:0] bits = 10b) | — |
| b5, b4 | SZ[1:0] | DTC Data Transfer Size | b5 b4<br>0 0: Byte (8-bit) transfer<br>0 1: Word (16-bit) transfer<br>1 0: Longword (32-bit) transfer<br>1 1: Setting prohibited | — |
| b7, b6 | MD[1:0] | DTC Transfer Mode Select | b7 b6<br>0 0: Normal transfer mode<br>0 1: Repeat transfer mode<br>1 0: Block transfer mode<br>1 1: Setting prohibited | — |

MRA cannot be accessed directly from the CPU.

## 16.2.2 DTC Mode Register B (MRB)

Address(es): (inaccessible directly from the CPU)

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | CHNE | CHNS | DISEL | DTS | DM[1:0] | | — | — |
| Value after reset: | x | x | x | x | x | x | x | x |

x: Undefined

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b1, b0 | — | Reserved | These bits are read as undefined. The write value should be 0. | — |
| b3, b2 | DM[1:0] | Transfer Destination Address Addressing Mode | b3 b2<br>0 0: Address in the DAR register is fixed. (Write-back to DAR is skipped.)<br>0 1: Address in the DAR register is fixed. (Write-back to DAR is skipped.)<br>1 0: DAR value is incremented after data transfer. (+1 when SZ[1:0] bits in MRA = 00b, +2 when SZ[1:0] bits = 01b, +4 when SZ[1:0] bits = 10b)<br>1 1: DAR value is decremented after data transfer. (–1 when SZ[1:0] bits in MRA = 00b, –2 when SZ[1:0] bits = 01b, –4 when SZ[1:0] bits = 10b) | — |
| b4 | DTS | DTC Transfer Mode Select | 0: Transfer destination side is repeat area or block area.<br>1: Transfer source side is repeat area or block area. | — |
| b5 | DISEL | DTC Interrupt Select | 0: An interrupt request to the CPU is generated when specified data transfer is completed.<br>1: An interrupt request to the CPU is generated each time DTC data transfer is performed. | — |
| b6 | CHNS | DTC Chain Transfer Select | 0: Chain transfer is performed continuously.<br>1: Chain transfer is performed only when the transfer counter is changed from 1 to 0 or 1 to CRAH. | — |
| b7 | CHNE | DTC Chain Transfer Enable | 0: Chain transfer is disabled.<br>1: Chain transfer is enabled. | — |

MRB cannot be accessed directly from the CPU.

### DTS Bit (DTC Transfer Mode Select)

The DTS bit specifies the side (transfer source or destination) to be a repeat area or block area in repeat transfer mode or block transfer mode.

### CHNS Bit (DTC Chain Transfer Select)

The CHNS bit selects the chain transfer condition.

When the CHNE bit is 0, setting of the CHNS bit is ignored. For details on the conditions to select the chain transfer, refer to Table 16.3, Chain Transfer Conditions.

When the next transfer is chain transfer, completion of the specified number of transfers is not determined, the startup source flag is not cleared, and an interrupt request to the CPU is not generated.

### CHNE Bit (DTC Chain Transfer Enable)

The CHNE bit enables or disables chain transfer.

The chain transfer condition is selected by the CHNS bit.

For details of chain transfer, refer to section 16.4.6, Chain Transfer.

## 16.2.3 DTC Transfer Source Register (SAR)

Address(es): (inaccessible directly from the CPU)

| | b31 | b30 | b29 | b28 | b27 | b26 | b25 | b24 | b23 | b22 | b21 | b20 | b19 | b18 | b17 | b16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Value after reset: | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x |

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Value after reset: | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x |

x: Undefined

SAR is used to set the transfer source start address.
In full-address mode, 32 bits are valid.
In short-address mode, lower 24 bits are valid and upper 8 bits (b31 to b24) are ignored. The address of this register is extended by the value specified by b23.
SAR cannot be accessed directly from the CPU.

## 16.2.4 DTC Transfer Destination Register (DAR)

Address(es): (inaccessible directly from the CPU)

| | b31 | b30 | b29 | b28 | b27 | b26 | b25 | b24 | b23 | b22 | b21 | b20 | b19 | b18 | b17 | b16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Value after reset: | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x |

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Value after reset: | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x |

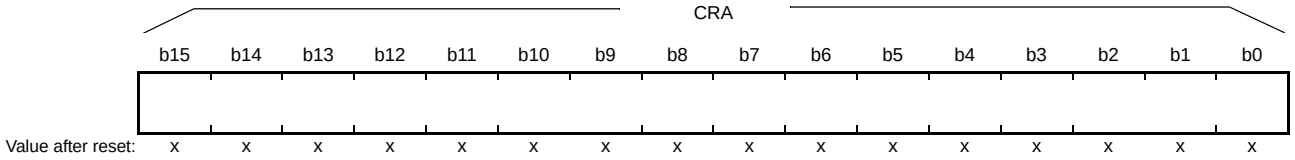
x: Undefined

DAR is used to set the transfer destination start address.
In full-address mode, 32 bits are valid.
In short-address mode, lower 24 bits are valid and upper 8 bits (b31 to b24) are ignored. The address of this register is extended by the value specified by b23.
DAR cannot be accessed directly from the CPU.

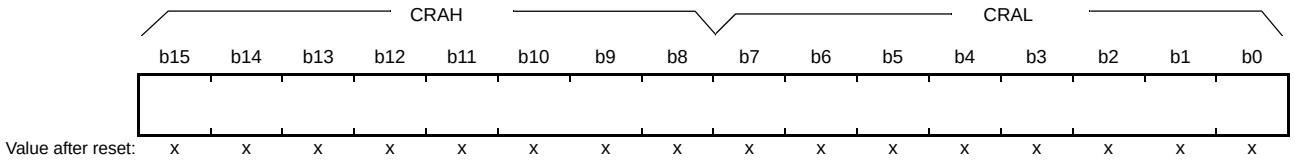
## 16.2.5 DTC Transfer Count Register A (CRA)

Address(es): (inaccessible directly from the CPU)

- Normal transfer mode




- Repeat transfer mode/block transfer mode




x: Undefined

Note: • The function depends on transfer mode.

| Symbol | Register Name | Description | R/W |
|---|---|---|---|
| CRAL | Transfer Counter A Lower Register | Set transfer count. | — |
| CRAH | Transfer Counter A Upper Register | | — |

Note: • Set CRAH and CRAL to the same value in repeat transfer mode and block transfer mode.

CRA cannot be accessed directly from the CPU.

### (1) Normal transfer mode (MRA.MD[1:0] bits = 00b)

CRA functions as a 16-bit transfer counter in normal transfer mode.
The transfer count is 1, 65535, and 65536 when the set value is 0001h, FFFFh, and 0000h, respectively.
The CRA value is decremented (–1) at each data transfer.

### (2) Repeat transfer mode (MRA.MD[1:0] bits = 01b)

The CRAH register retains transfer count and the CRAL register functions as an 8-bit transfer counter.
The transfer count is 1, 255, and 256 when the set value is 01h, FFh, and 00h, respectively.
The CRAL value is decremented (–1) at each data transfer. When it reaches 00h, the CRAH value is transferred to CRAL.

### (3) Block transfer mode (MRA.MD[1:0] bits = 10b)

The CRAH register retains block size and the CRAL register functions as an 8-bit block size counter.
The transfer count is 1, 255, and 256 when the set value is 01h, FFh, and 00h, respectively.
The CRAL value is decremented (–1) at each data transfer. When it reaches 00h, the CRAH value is transferred to CRAL.

## 16.2.6 DTC Transfer Count Register B (CRB)

Address(es): (inaccessible directly from the CPU)

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Value after reset: | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x |

x: Undefined

CRB is used to set the block transfer count for block transfer mode.
The transfer count is 1, 65535, and 65536 when the set value is 0001h, FFFFh, and 0000h, respectively. The CRB value is decremented (–1) when the final data of a single block size is transferred.
When normal transfer mode or repeat transfer mode is selected, this register is not used and the set value is ignored.
CRB cannot be accessed directly from the CPU.

## 16.2.7 DTC Control Register (DTCCR)

Address(es): 0008 2400h

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | — | RRS | — | — | — | — |
| Value after reset: | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b2 to b0 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |
| b3 | — | Reserved | This bit is read as 1. The write value should be 1. | R/W |
| b4 | RRS | DTC Transfer Information Read Skip Enable | 0: Transfer information read is not skipped.<br>1: Transfer information read is skipped when vector numbers match. | R/W |
| b7 to b5 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |

**RRS Bit (DTC Transfer Information Read Skip Enable)**

The DTC vector number is always compared with the vector number in the previous startup process.
When these vector numbers match and the RRS bit is set to 1, DTC data transfer is performed without reading the transferred information. However, when the previous transfer was chain transfer, the transferred information is always read regardless of the value of the RRS bit.
Furthermore, when the transfer counter (CRA register) became 0 during the previous normal transfer and when the transfer counter (CRB register) became 0 during the previous block transfer, the transferred information is always read regardless of the value of the RRS bit.

## 16.2.8 DTC Vector Base Register (DTCVBR)

Address(es): 0008 2404h

| | b31 | b30 | b29 | b28 | b27 | b26 | b25 | b24 | b23 | b22 | b21 | b20 | b19 | b18 | b17 | b16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Bit Name | Description | R/W |
|---|---|---|---|
| b9 to b0 | DTC Vector Base Address (Lower 10 bits) | These bits are read as 0. The write value should be 0. | R |
| b31 to b10 | DTC Vector Base Address (Upper 22 bits) | Writing to the upper 4 bits (b31 to b28) is ignored, and the address of this register is extended by the value specified by b27. | R/W |

DTCVBR is used to set the base address for calculating the DTC vector table address.
It can be set in the range of 0000 0000h to 07FF FC00h and F800 0000h to FFFF FC00h in 1-Kbyte units.

## 16.2.9 DTC Address Mode Register (DTCADMOD)

Address(es): 0008 2408h

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | SHORT |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b0 | SHORT | Short-Address Mode | 0: Full-address mode<br>1: Short-address mode | R/W |
| b7 to b1 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |

DTCADMOD is used to specify the area accessible by the DTC.

### SHORT Bit (Short-Address Mode)

Full-address mode allows the DTC to access to a 4-Gbyte space (0000 0000h to FFFF FFFFh).
Short-address mode allows the DTC to access to a 16-Mbyte space (0000 0000h to 007F FFFFh and FF80 0000h to FFFF FFFFh).

## 16.2.10 DTC Module Start Register (DTCST)

Address(es): 0008 240Ch

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | DTCST |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b0 | DTCST | DTC Module Start | 0: DTC module stop<br>1: DTC module start | R/W |
| b7 to b1 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |

**DTCST Bit (DTC Module Start)**

Set the DTCST bit to 1 to enable the DTC to accept transfer requests. When this bit is cleared to 0, transfer requests are no longer accepted.

If this bit is cleared to 0 during data transfer, the accepted transfer request is active until the processing is completed.

Before making transition to the module stop state or software standby mode, the DTCST bit must be set to 0.

For details on transitions to the module stop state and software standby mode, refer to section 16.9, Low Power Consumption Function, and section 11, Low Power Consumption.

## 16.2.11 DTC Status Register (DTCSTS)

Address(es): 0008 240Eh

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | ACT | — | — | — | — | — | — | — | VECN[7:0] | | | | | | | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b7 to b0 | VECN[7:0] | DTC-Activating Vector Number Monitoring | These bits indicate the vector number for the activating source when DTC transfer is in progress.<br>The value is only valid if DTC transfer is in progress (the value of the ACT flag is 1). | R |
| b14 to b8 | — | Reserved | These bits are read as 0. Writing to this bit has no effect. | R |
| b15 | ACT | DTC Active Flag | 0: DTC transfer operation is not in progress.<br>1: DTC transfer operation is in progress. | R |

### VECN[7:0] Bits (DTC-Activating Vector Number Monitoring)

While transfer by the DTC is in progress, these bits indicate the vector number corresponding to the activating source for the transfer.

When the DTCSTS register is read, the value read from the VECN[7:0] bits is valid if the value of the ACT flag was 1 (indicating DTC transfer in progress) and invalid if the value of the ACT flag was 0 (indicating no current DTC transfer).

For the correspondence between the DTC startup sources and the vector addresses, refer to Table 14.3, Interrupt Vector Table in section 14, Interrupt Controller (ICUb).

### ACT Flag (DTC Active Flag)

This bit indicates the state of DTC transfer operation.

[Setting condition]

- When the DTC is activated by a transfer request.

[Clearing condition]

- When transfer by the DTC is completed in response to a transfer request.

## 16.3 Sources of Activation

The DTC is activated by an interrupt request. Setting the DTCERn.DTCE bit (where n is the interrupt vector number of the given interrupt) of the ICU to 1 selects the corresponding interrupt as an activation source for the DTC.
For the correspondence between the DTC startup sources and the vector addresses, refer to Table 14.3, Interrupt Vector Table in section 14, Interrupt Controller (ICUb). For startup by software, refer to section 14.2.5, Software Interrupt Activation Register (SWINTR) in section 14, Interrupt Controller (ICUb).
Once the DTC has accepted a startup request, it does not accept another startup request until transfer for that single request is completed, regardless of the priority of the requests. When multiple startup requests are generated during DTC transfer, a request with the highest priority on completion of the transfer is accepted. When multiple startup requests are generated while the DTC module start bit (DTCST.DTCST) is 0, a request with the highest priority at the moment when the bit is subsequently set to 1 is accepted.

The DTC performs the following operations at the start of a single data transfer (or the last of the consecutive transfers in the case of a chained transfer).

- On completion of a specified round of data transfer, the DTCERn.DTCE bit is cleared to 0 and an interrupt is requested to the CPU.
- If the MRB.DISEL bit is 1, an interrupt is requested to the CPU on completion of data transfer.
- For the other transfers, the interrupt status flag of the startup source is cleared to 0 at the start of data transfer.

### 16.3.1 Allocating Transfer Information and DTC Vector Table

The DTC reads the start address of the transfer information corresponding to each startup source from the vector table and reads the transfer information starting at that address.
The vector table should be located so that the lower 10 bits of the base address (start address) are 0. Use the DTC vector base register (DTCVBR) to set the base address of the DTC vector table.
Transfer information is allocated in the RAM area. In the RAM area, the start address of the transfer information (n) with vector number n should be 4n added to the base address in the vector table.
Transfer information can be allocated in short-address mode (3 longwords) or full-address mode (4 longwords). Use the DTCADMOD.SHORT bit to select short-address mode (SHORT bit = 1) or full-address mode (SHORT bit = 0).
Figure 16.2 shows the relationship between the DTC vector table and transfer information.
Figure 16.3 shows the allocation of transfer information in the RAM area. The lower addresses vary according to the endian of the corresponding allocation area. For details, refer to section 16.10.2, Allocating Transfer Information.

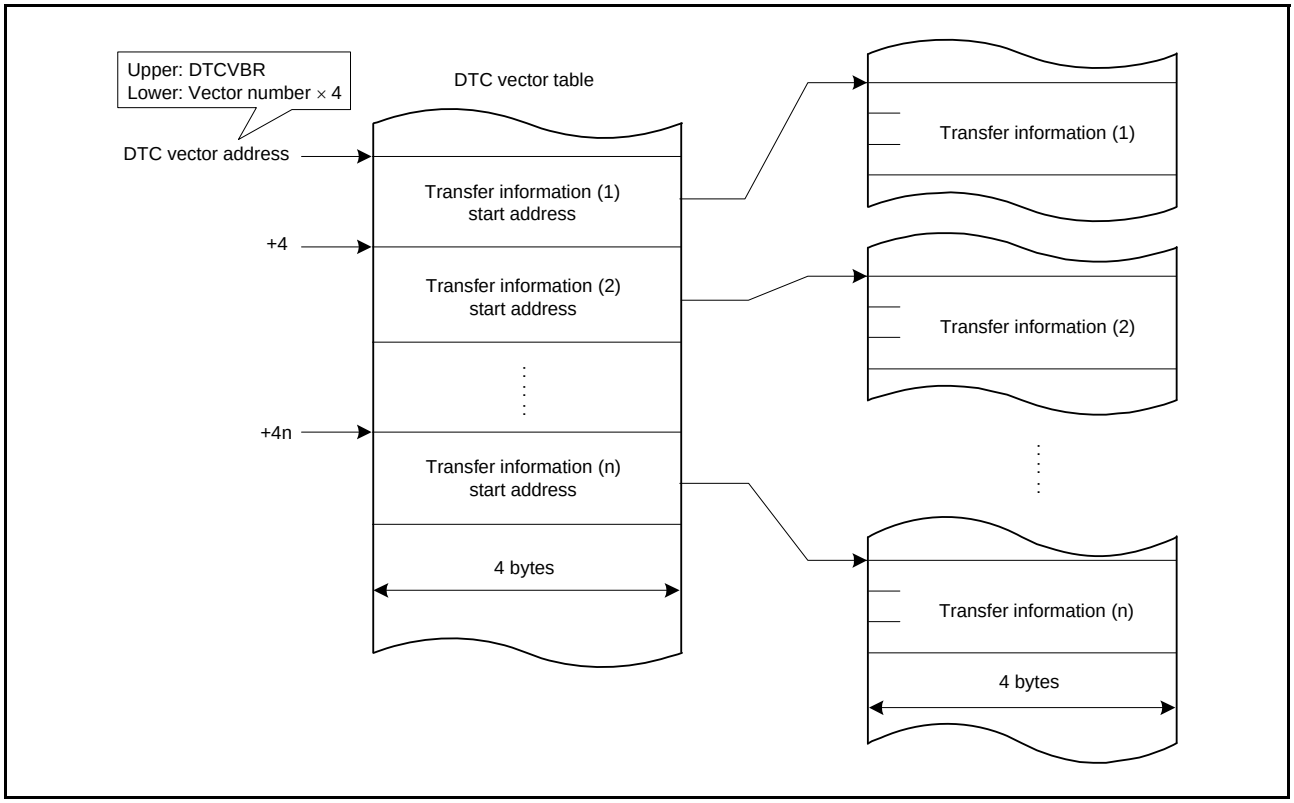



**Figure 16.2 DTC Vector Table and Transfer Information**

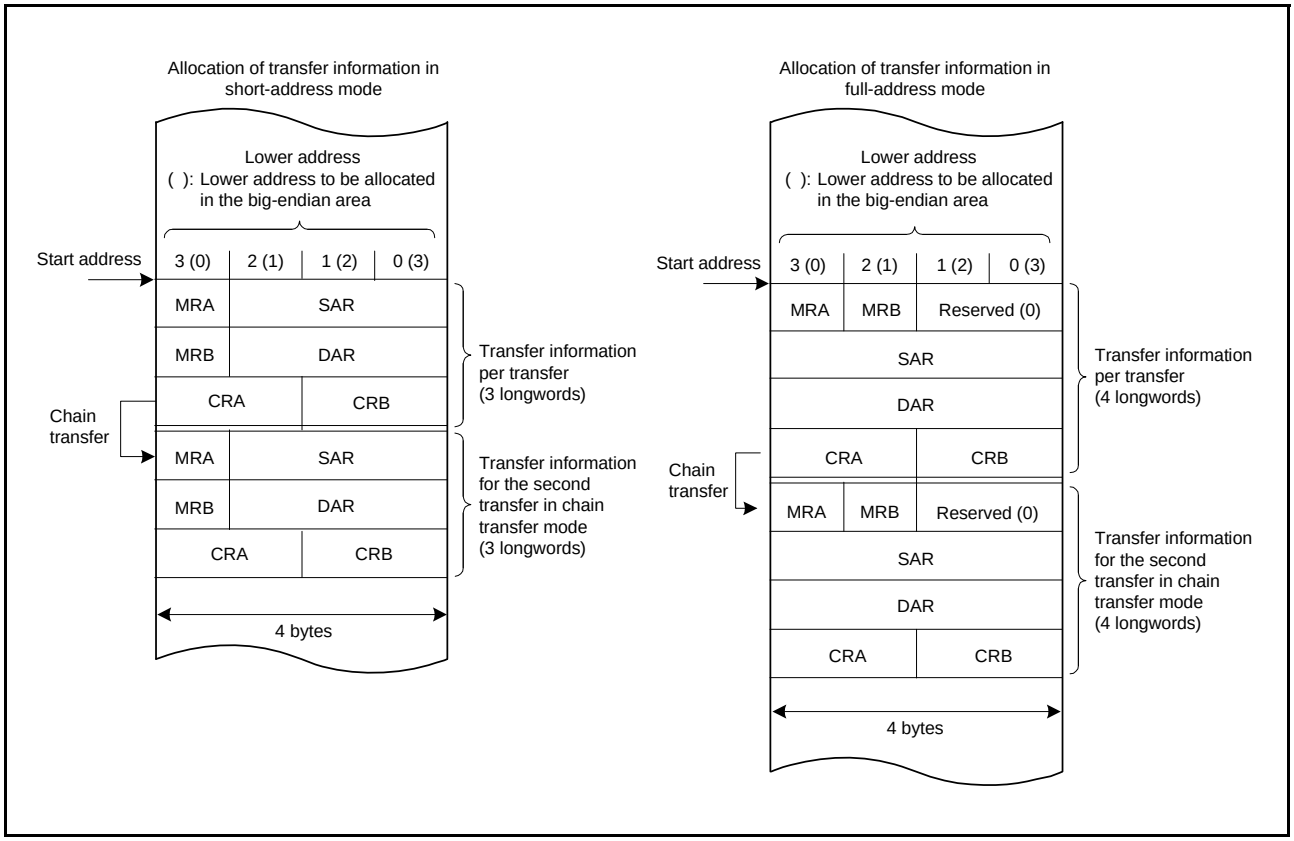



**Figure 16.3 Allocation of Transfer Information in the RAM Area**

## 16.4 Operation

The DTC transfers data in accordance with the transfer information. Storage of the transfer information in the RAM area is required before DTC operation.

When the DTC is activated, it reads the DTC vector corresponding to the vector number. Then the DTC reads transfer information from the transfer information store address pointed by the DTC vector, transfers data, and then writes back the transfer information after the data transfer. Storing transfer information in the RAM area allows data transfer of arbitrary number of channels.

There are three transfer modes: normal transfer mode, repeat transfer mode, and block transfer mode.

The DTC specifies a transfer source address in the SAR register and a transfer destination address in the DAR register. The values of these registers are incremented, decremented, or address-fixed independently after data transfer.

Table 16.2 lists transfer modes of the DTC.

**Table 16.2 Transfer Modes of the DTC**

| Transfer Mode | Data Size Transferred on a Single Transfer Request | Increment/Decrement of Memory Address | Settable Transfer Count |
|---|---|---|---|
| Normal transfer mode | 1 byte/1 word/1 longword | Incremented/decremented by 1, 2, or 4 or address fixed | 1 to 65536 |
| Repeat transfer mode*1 | 1 byte/1 word/1 longword | Incremented/decremented by 1, 2, or 4 or address fixed | 1 to 256*3 |
| Block transfer mode*2 | Block size specified in CRAH (1 to 256 bytes/1 to 256 words/1 to 256 longwords) | Incremented/decremented by 1, 2, or 4 or address fixed | 1 to 65536 |

Note 1. Set transfer source or transfer destination in the repeat area.
Note 2. Set transfer source or transfer destination in the block area.
Note 3. After data transfer of the specified count, the initial state is restored and the operation is continued (repeated).

Setting the MRB.CHNE bit to 1 allows multiple transfers (chain transfer) on a single startup source. Setting the MRB.CHNS bit also enables chain transfer when specified data transfer is completed.

Figure 16.4 shows the operation flowchart of the DTC. Table 16.3 lists chain transfer conditions. (The combination of control information for the second and third transfers and that for transfers following the third are omitted in this table).

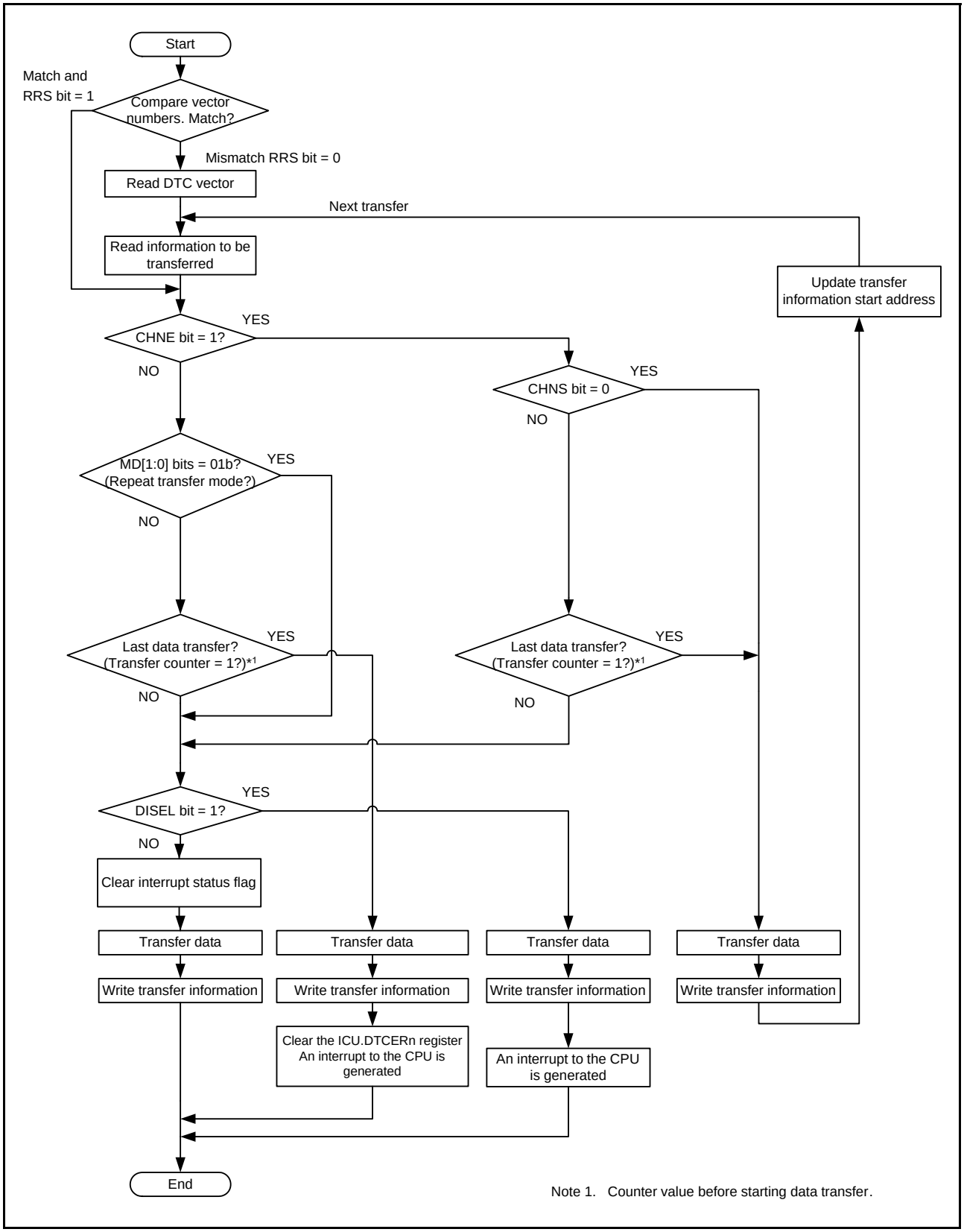



**Figure 16.4 Operation Flowchart of the DTC**

**Table 16.3 Chain Transfer Conditions**

| First Transfer | | | | Second Transfer*3 | | | | |
|---|---|---|---|---|---|---|---|---|
| CHNE Bit | CHNS Bit | DISEL Bit | Transfer Counter*1,*2 | CHNE Bit | CHNS Bit | DISEL Bit | Transfer Counter*1,*2 | DTC Transfer |
| 0 | — | 0 | Other than (1 → 0) | — | — | — | — | Ends after the first transfer |
| 0 | — | 0 | (1 → 0) | — | — | — | — | Ends after the first transfer with an interrupt request to the CPU |
| 0 | — | 1 | — | — | — | — | — | |
| 1 | 0 | — | — | 0 | — | 0 | Other than (1 → 0) | Ends after the second transfer |
| | | | | 0 | — | 0 | (1 → 0) | Ends after the second transfer with an interrupt request to the CPU |
| | | | | 0 | — | 1 | — | |
| 1 | 1 | 0 | Other than (1 → *) | — | — | — | — | Ends after the first transfer |
| 1 | 1 | — | (1 → *) | 0 | — | 0 | Other than (1 → 0) | Ends after the second transfer |
| | | | | 0 | — | 0 | (1 → 0) | Ends after the second transfer with an interrupt request to the CPU |
| | | | | 0 | — | 1 | — | |
| 1 | 1 | 1 | Other than (1 → *) | — | — | — | — | Ends after the first transfer with an interrupt request to the CPU |

Note 1. The transfer counters used depend on transfer modes as follows:
Normal transfer mode: CRA register
Repeat transfer mode: CRAL register
Block transfer mode: CRB register

Note 2. On completion of data transfer, the counters operate as follows:
1 → 0: in normal and block transfer modes
1 → CRAH: in repeat transfer mode
(1 → *) in the table indicates both of the two operations above.

Note 3. Chain transfer can be selected for the second or subsequent transfers. The condition combination of “second transfer and CHNE bit = 1” is omitted.

### 16.4.1 Transfer Information Read Skip Function

Reading of vector addresses and transfer information can be skipped by the setting of the DTCCR.RRS bit.

When a DTC startup request is generated, the current DTC vector number is always compared with the DTC vector number in the previous startup process. When these vector numbers match and the RRS bit is set to 1, DTC data transfer is performed without reading the vector address and transfer information. However, when the previous transfer was chain transfer, the vector address and transfer information are always read. Furthermore, when the transfer counter (CRA register) became 0 during the previous normal transfer and when the transfer counter (CRB register) became 0 during the previous block transfer, transfer information is always read regardless of the value of the RRS bit. Figure 16.13 shows an example of transfer information read skip.

To update the vector table and transfer information, set the RRS bit to 0, update the vector table and transfer information, and then set the RRS bit to 1. When the RRS bit is set to 0, the retained vector number is discarded and the vector table and transfer information that are updated in the following startup process are read.

### 16.4.2 Transfer Information Write-Back Skip Function

When the MRA.SM[1:0] bits or the MRB.DM[1:0] bits are set to “address fixed”, a part of transfer information is not written back. This function is performed independently of the setting of short-address mode or full-address mode. Table 16.4 lists transfer information write-back skip conditions and applicable registers.
The CRA and CRB registers are always written back independently of the setting of short-address mode or full-address mode. Furthermore, in full-address mode, write-back of the MRA and MRB registers are always skipped.

**Table 16.4 Transfer Information Write-Back Skip Conditions and Applicable Registers**

| **MRA.SM[1:0] Bits** | | **MRB.DM[1:0] Bits** | | | |
|---|---|---|---|---|---|
| b3 | b2 | b3 | b2 | **SAR Register** | **DAR Register** |
| 0 | 0 | 0 | 0 | Skip | Skip |
| 0 | 0 | 0 | 1 | | |
| 0 | 1 | 0 | 0 | | |
| 0 | 1 | 0 | 1 | | |
| 0 | 0 | 1 | 0 | Skip | Write-back |
| 0 | 0 | 1 | 1 | | |
| 0 | 1 | 1 | 0 | | |
| 0 | 1 | 1 | 1 | | |
| 1 | 0 | 0 | 0 | Write-back | Skip |
| 1 | 0 | 0 | 1 | | |
| 1 | 1 | 0 | 0 | | |
| 1 | 1 | 0 | 1 | | |
| 1 | 0 | 1 | 0 | Write-back | Write-back |
| 1 | 0 | 1 | 1 | | |
| 1 | 1 | 1 | 0 | | |
| 1 | 1 | 1 | 1 | | |

### 16.4.3 Normal Transfer Mode

This mode allows 1-byte, 1-word, or 1-longword data transfer on a single startup source. The transfer count can be set to 1 to 65536.

Transfer source addresses and transfer destination addresses can be set to increment, decrement, or fixed independently.

This mode enables an interrupt request to the CPU to be generated at the end of specified-count transfer.

Table 16.5 lists register functions in normal transfer mode, and Figure 16.5 shows the memory map of normal transfer mode.

**Table 16.5 Register Functions in Normal Transfer Mode**

| Register | Description | Value Written Back by Writing Transfer Information |
|---|---|---|
| SAR | Transfer source address | Increment/decrement/fixed*1 |
| DAR | Transfer destination address | Increment/decrement/fixed*1 |
| CRA | Transfer counter A | CRA - 1 |
| CRB | Transfer counter B | Not updated |

Note 1. Write-back operation is skipped in address-fixed mode.

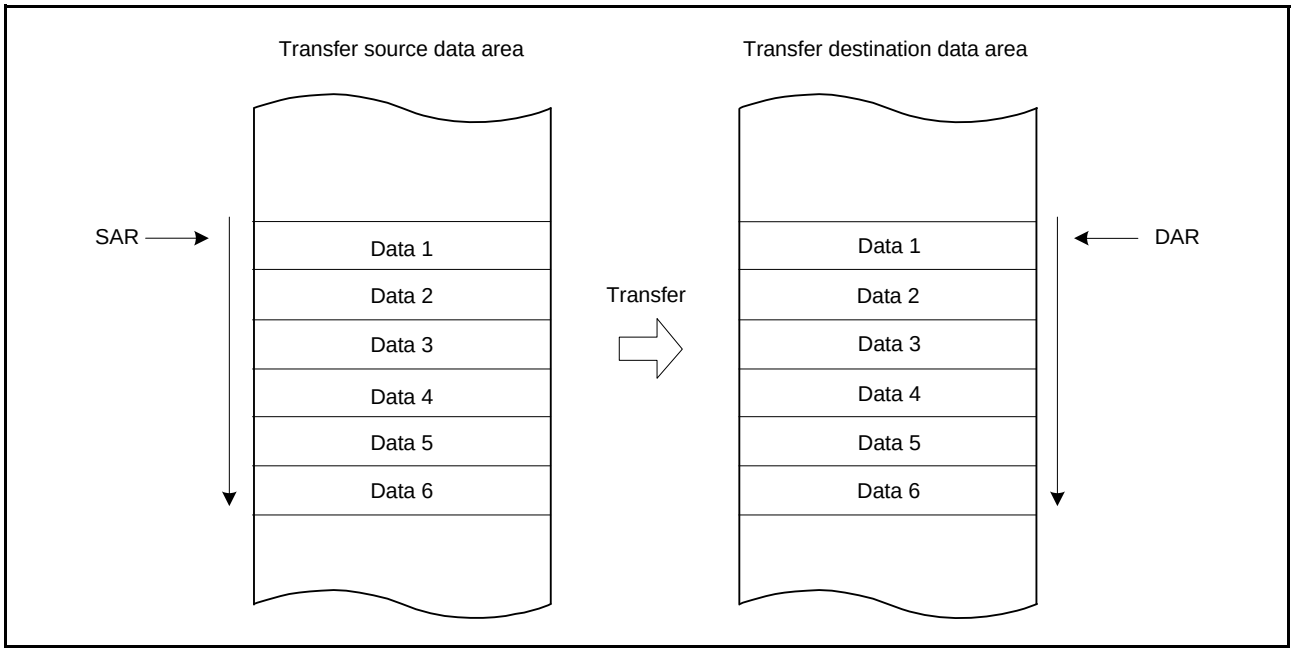



**Figure 16.5 Memory Map of Normal Transfer Mode**

### 16.4.4 Repeat Transfer Mode

This mode allows 1-byte, 1-word, or 1-longword data transfer on a single startup source.
Specify either transfer source or transfer destination for the repeat area by the MRB.DTS bit. The transfer count can be set to 1 to 256. When the specified-count transfer is completed, the initial value of the address register specified in the transfer counter and the repeat area is restored and transfer is repeated. The other address register is incremented or decremented continuously or remains unchanged.
When the transfer counter CRAL is decreased to 00h in repeat transfer mode, the CRAL value is updated to the value set in the CRAH register. Thus the transfer counter does not become 00h, which inhibits generation of interrupt request to the CPU when the MRB.DISEL bit is set to 0 (an interrupt request to the CPU is generated when specified data transfer is completed).
Table 16.6 lists the register functions in repeat transfer mode, and Figure 16.6 shows the memory map of repeat transfer mode.

**Table 16.6 Register Functions in Repeat Transfer Mode**

| | | Value Written Back by Writing Transfer Information | |
|---|---|---|---|
| **Register** | **Description** | **When CRAL is not 1** | **When CRAL is 1** |
| SAR | Transfer source address | Increment/decrement/fixed*1 | (When the MRB.DTS bit is 0)<br>Increment/decrement/fixed*1<br>(When the MRB.DTS bit is 1)<br>SAR register initial value |
| DAR | Transfer destination address | Increment/decrement/fixed*1 | (When the MRB.DTS bit is 0)<br>DAR register initial value<br>(When the MRB.DTS bit is 1)<br>Increment/decrement/fixed*1 |
| CRAH | Retains transfer counter | CRAH | CRAH |
| CRAL | Transfer counter A | CRAL - 1 | CRAH |
| CRB | Transfer counter B | Not updated | Not updated |

Note 1. Write-back is skipped in address-fixed mode.

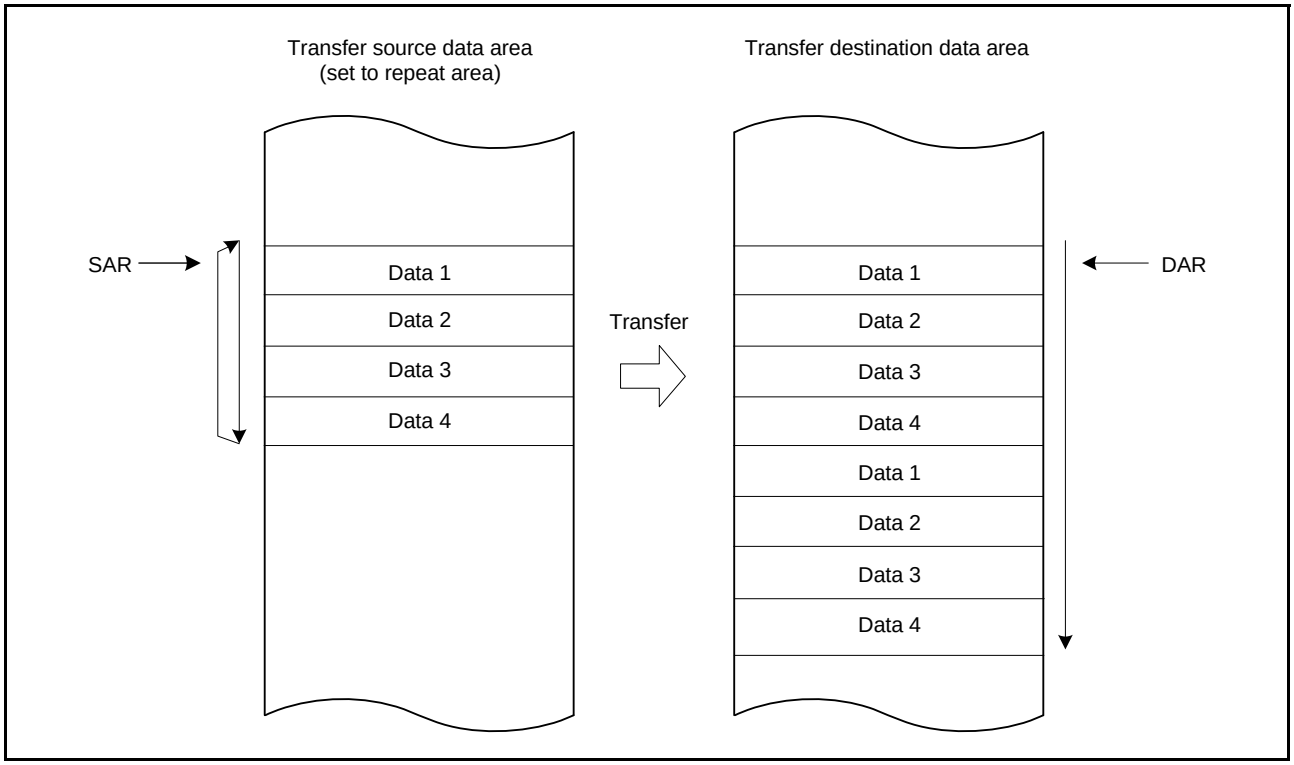



**Figure 16.6 Memory Map of Repeat Transfer Mode (Transfer Source: Repeat Area)**

### 16.4.5 Block Transfer Mode

This mode allows single-block data transfer on a single startup source.
Specify either transfer source or transfer destination for the block area by the MRB.DTS bit. The block size can be set to 1 to 256 bytes, 1 to 256 words, or 1 to 256 longwords.
When transfer of the specified one block is completed, the initial values of the block size counter CRAL and the address register (the SAR register when the MRB.DTS bit = 1 or the DAR register when the DTS bit = 0) specified in the block area are restored. The other address register is incremented or decremented continuously or remains unchanged.
The transfer count (block count) can be set to 1 to 65536. This mode enables an interrupt request to the CPU to be generated at the end of specified-count block transfer.
Table 16.7 lists register functions in block transfer mode, and Figure 16.7 shows the memory map of block transfer mode.

**Table 16.7 Register Functions in Block Transfer Mode**

| Register | Description | Value Written Back by Writing Transfer Information |
|---|---|---|
| SAR | Transfer source address | (When MRB.DTS bit is 0)<br>Increment/decrement/fixed*1<br>(When MRB.DTS bit is 1)<br>SAR register initial value |
| DAR | Transfer destination address | (When MRB.DTS bit is 0)<br>DAR register initial value<br>(When MRB.DTS bit is 1)<br>Increment/decrement/fixed*1 |
| CRAH | Retains block size | CRAH |
| CRAL | Block size counter | CRAH |
| CRB | Block transfer counter | CRB - 1 |

Note 1. Write-back is skipped in address-fixed mode.

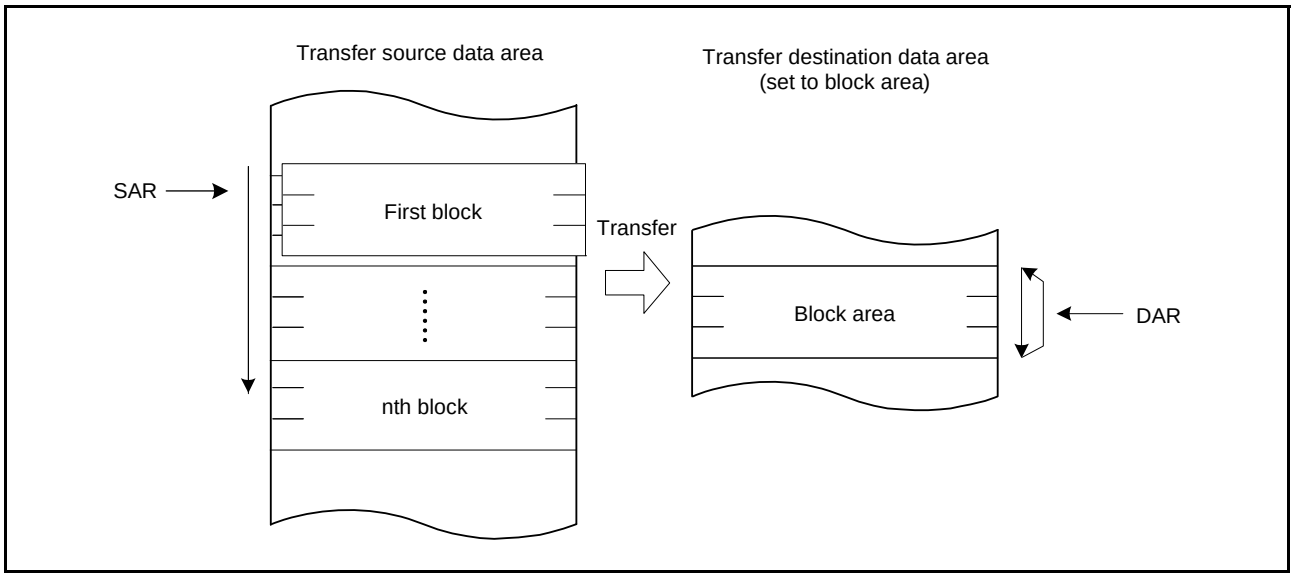



**Figure 16.7 Memory Map of Block Transfer Mode (Transfer Destination: Block Area)**

### 16.4.6 Chain Transfer

Setting the MRB.CHNE bit to 1 allows chain transfer to be performed continuously on a single startup source.
If the MRB.CHNE and CHNS bits are set to 1 and 0, respectively, an interrupt request to the CPU is not generated by completion of specified number of rounds of transfer or by setting the MRB.DISEL bit to 1 (an interrupt request to the CPU is generated each time DTC data transfer is performed), and data transfer has no effect on the interrupt status flag that has started up the transfer.
The SAR, DAR, CRA, CRB, MRA, and MRB registers can be set independently of each other to define data transfer.
Figure 16.8 shows chain transfer operation.

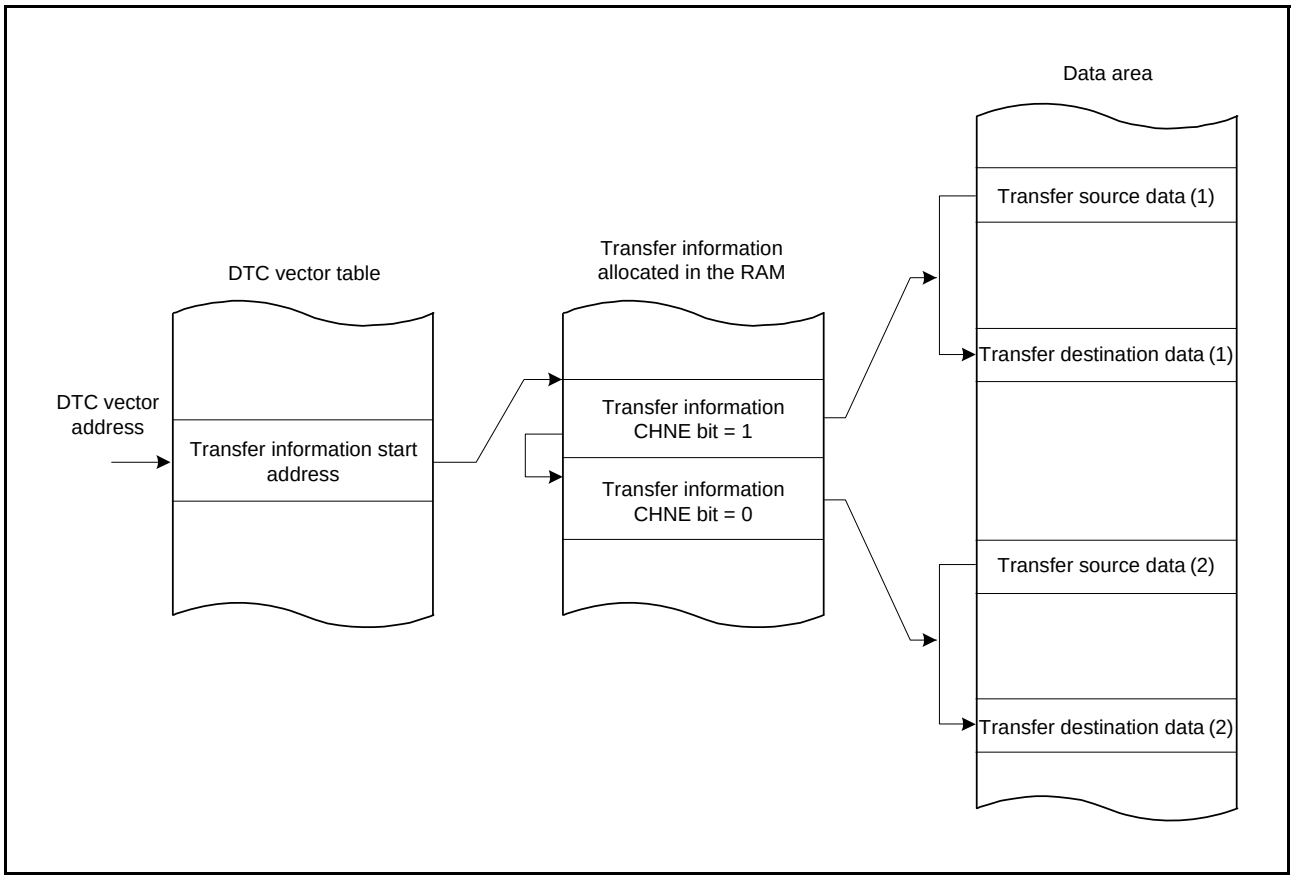



**Figure 16.8 Chain Transfer Operation**

Writing 1 to the MRB.CHNE and CHNS bits enables chain transfer to be performed only after completion of specified data transfer. In repeat transfer mode, chain transfer is performed after completion of specified data transfer.
For details on chain transfer conditions, see Table 16.3, Chain Transfer Conditions.

### 16.4.7 Operation Timing

Figure 16.9 to Figure 16.13 show examples of DTC operation timing.

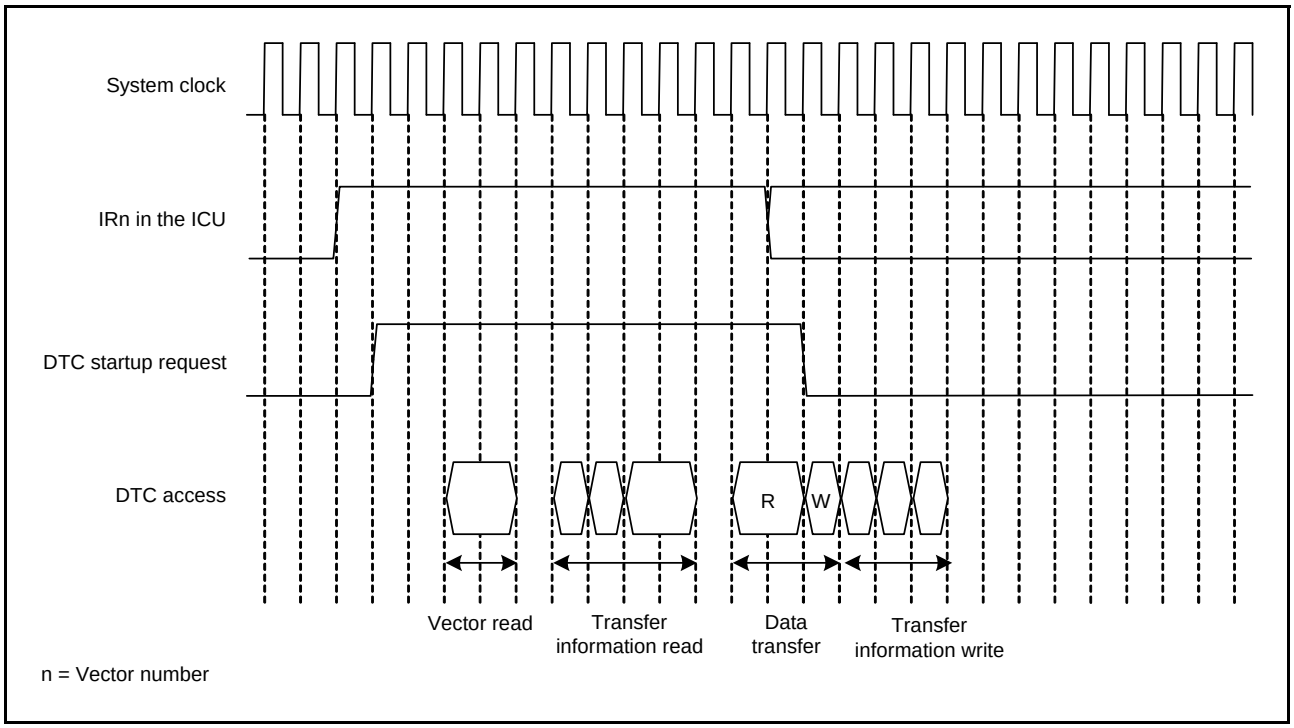



**Figure 16.9 Example (1) of DTC Operation Timing (Short-Address Mode, Normal Transfer Mode, Repeat Transfer Mode)**

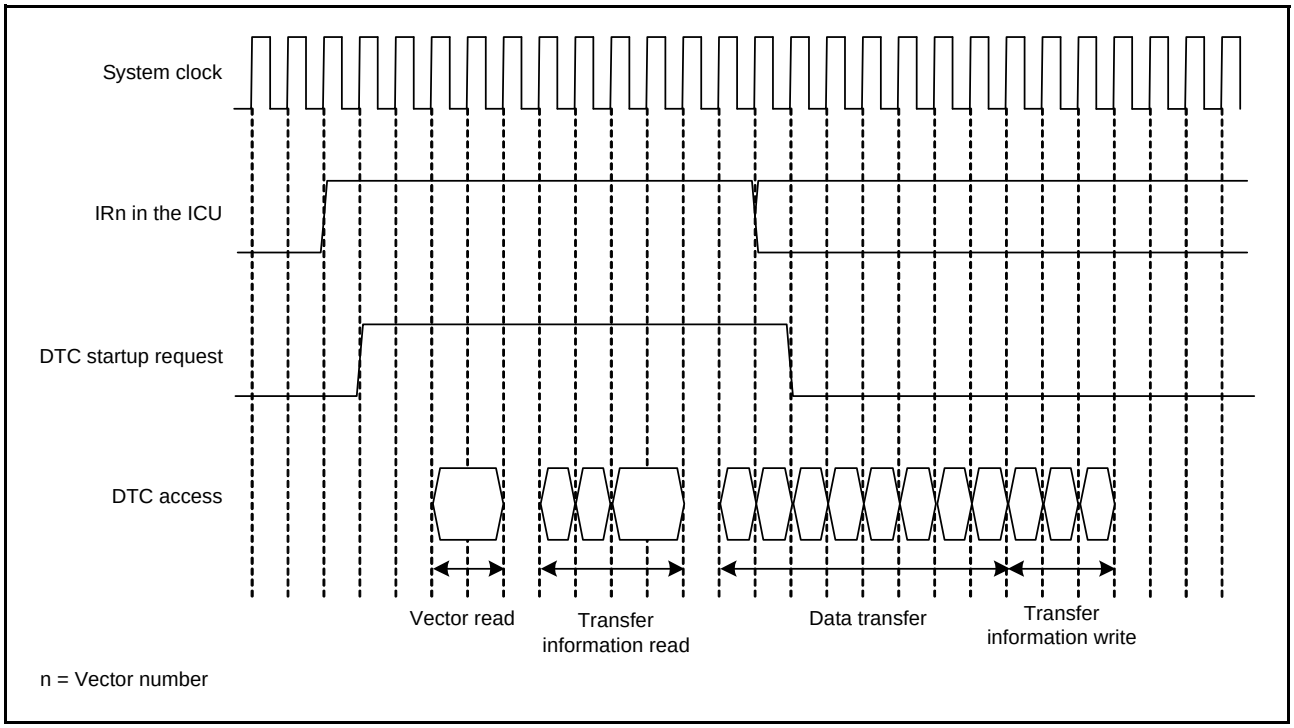



**Figure 16.10 Example (2) of DTC Operation Timing (Short-Address Mode, Block Transfer Mode, Block Size = 4)**

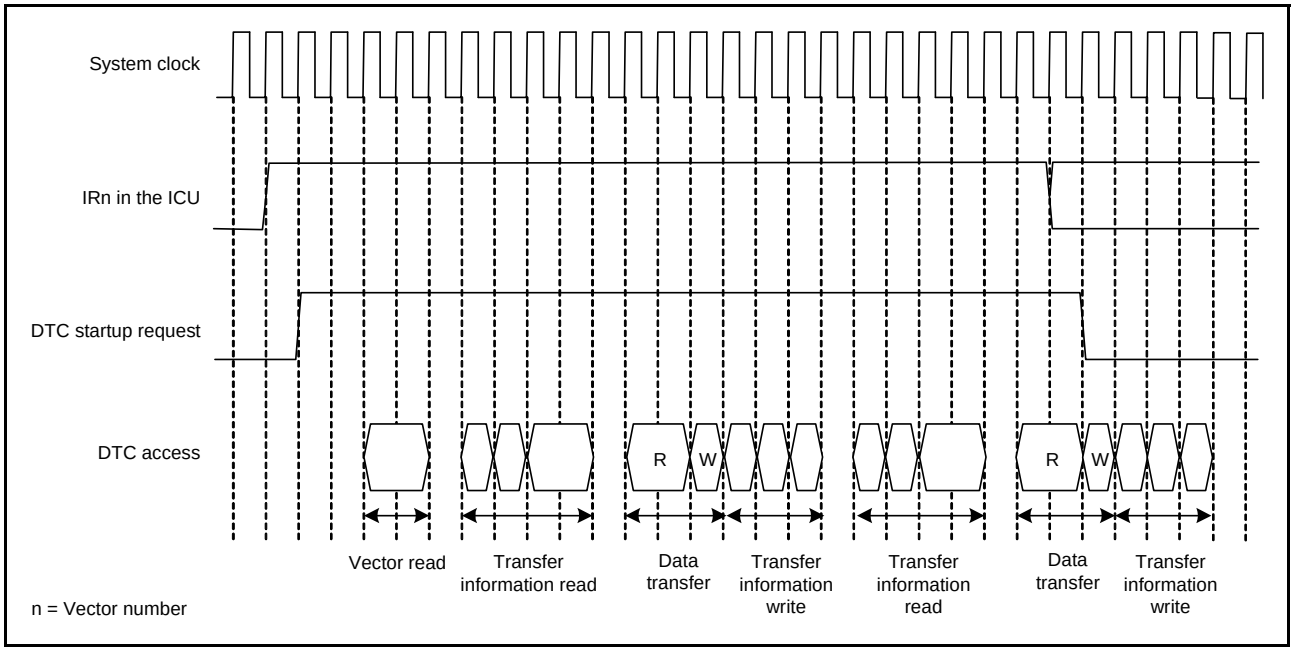



**Figure 16.11 Example (3) of DTC Operation Timing (Short-Address Mode, Chain Transfer)**

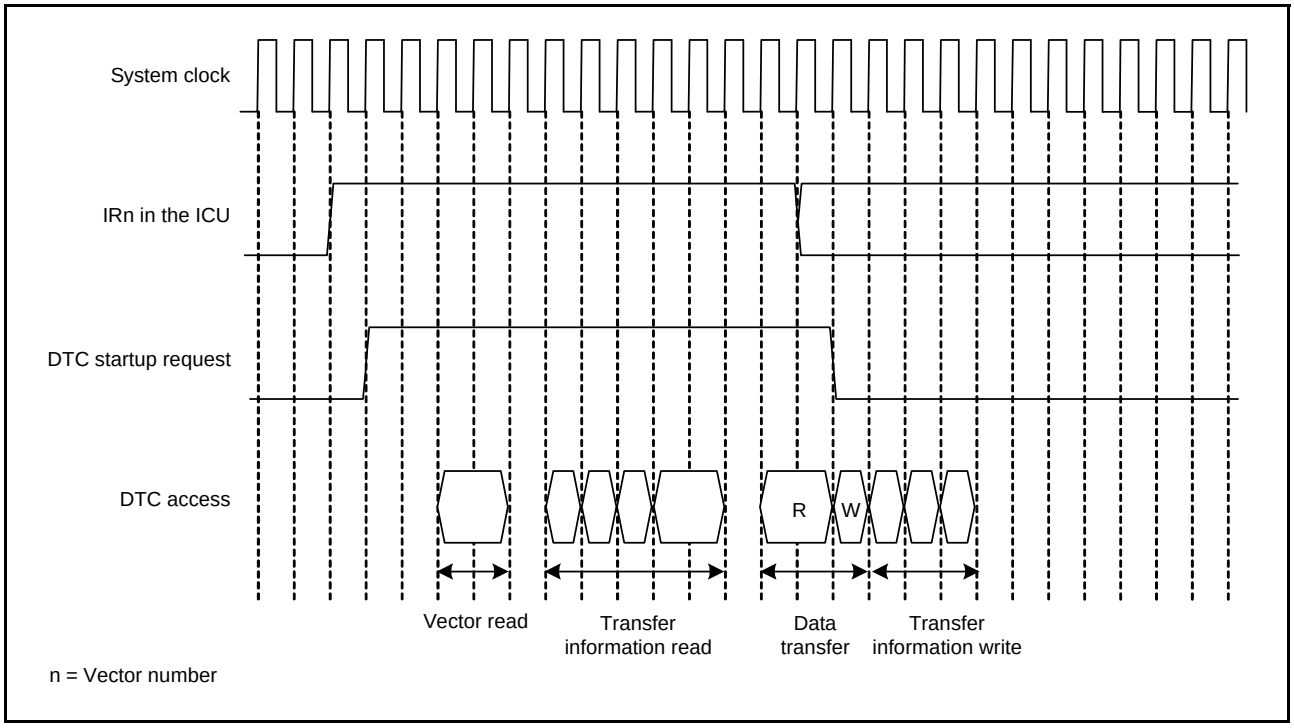
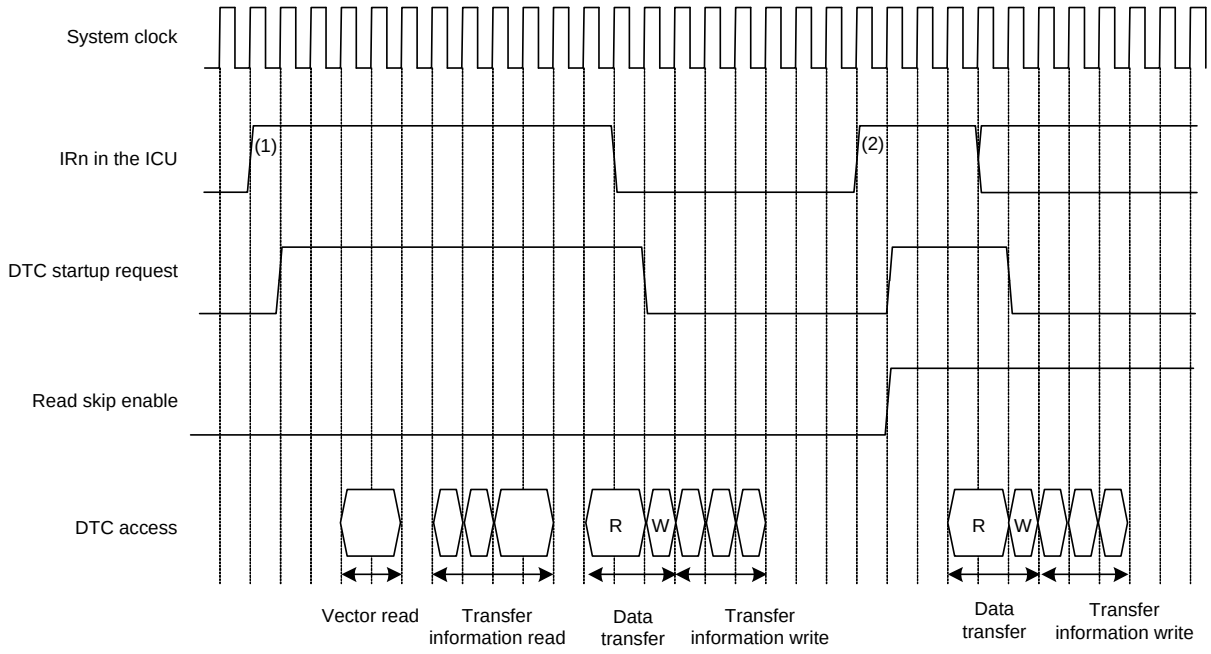



**Figure 16.12 Example (4) of DTC Operation Timing (Full-Address Mode, Normal Transfer Mode, Repeat Transfer Mode)**




n = Vector number

Note: • When startup sources (vector numbers) of (1) and (2) are the same and the RRS bit = 1, the transfer information read for request (2) is skipped.

**Figure 16.13 Example of Operation when Transfer Information Skip is Executed (Vector, Transfer Information, and Transfer Destination Data on the RAM, and Transfer Source Data on the Peripheral Module)**

## 16.4.8 Execution Cycles of the DTC

Table 16.8 lists the execution cycles of single data transfer of the DTC.
For the order of the execution states, refer to section 16.4.7, Operation Timing.

**Table 16.8 Execution Cycles of the DTC**

| Transfer Mode | Vector Read | | Transfer Information Read | | | Transfer Information Write | | | Data Transfer | | Internal Operation | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | | | Read | Write | | |
| Normal | Cv+1 | 0*1 | 4×Ci+1*2 | 3×Ci+1*3 | 0*1 | 3×Ci*4 | 2×Ci*5 | Ci*6 | Cr+1 | Cw | 2 | 0*1 |
| Repeat | | | | | | | | | Cr+1 | Cw | | |
| Block*7 | | | | | | | | | P×Cr | P×Cw | | |

Note 1. When transfer information read is skipped
Note 2. In full-address mode
Note 3. In short-address mode
Note 4. When neither SAR nor DAR is set to address-fixed mode
Note 5. When SAR or DAR is set to address-fixed mode
Note 6. When SAR and DAR are set to address-fixed mode
Note 7. When the block size is 2 or more. If the block size is 1, the cycle number for normal transfer is applied.

P: Block size (initial settings of CRAH and CRAL)
Cv: Cycles for access to vector transfer information storage destination
Ci: Cycles for access to transfer information storage destination address
Cr: Cycles for access to data read destination
Cw: Cycles for access to data write destination
(The unit is system clocks (ICLK) for “+ 1” in the Vector Read, Transfer Information Read, and Data Transfer Read columns and “2” in the Internal Operation column.)
(Cv, Ci, Cr, and Cw vary depending on the corresponding access destination. For the number of cycles for respective access destinations, refer to section 34, RAM, section 35, Flash Memory, and section 5, I/O Registers.)

## 16.4.9 DTC Bus Mastership Release Timing

The DTC does not release the bus mastership during transfer information read and transfer information write. While transfer information is not read or written, bus arbitration is made according to the priority determined by the bus master arbitrator.
For bus arbitration, refer to section 15, Buses.

## 16.5 DTC Setting Procedure

Before using the DTC, set the DTC vector base register (DTCVBR).
Figure 16.14 shows the procedure to set the DTC.

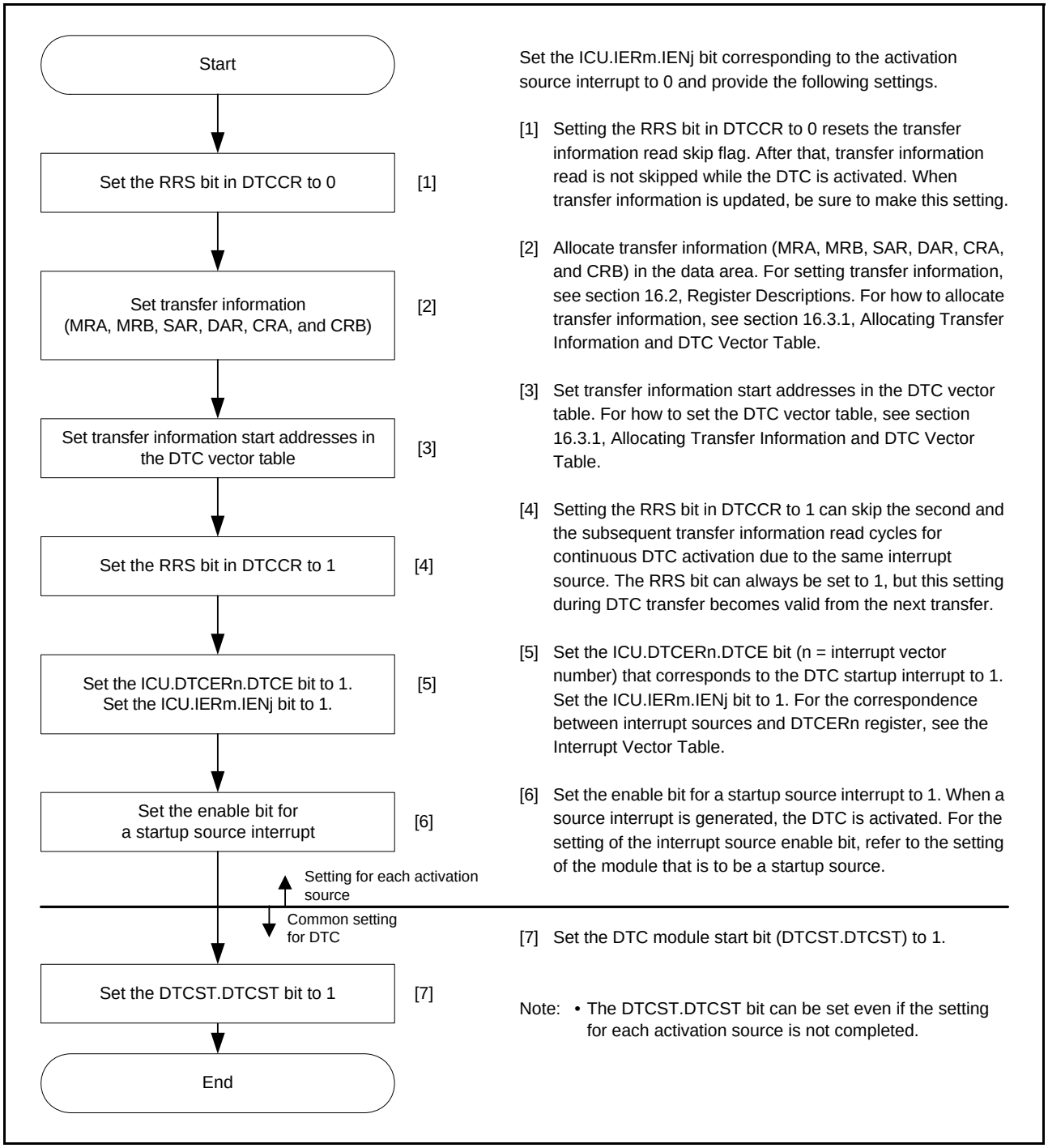



**Figure 16.14 Procedure to Set the DTC**

## 16.6 Examples of DTC Usage

### 16.6.1 Normal Transfer

As an example of DTC usage, its employment in the reception of 128 bytes of data by an SCI is described below.

#### (1) Transfer Information Set

In the MRA register, select a fixed source address (MRA.SM[1:0] bits = 00b), normal transfer mode (MRA.MD[1:0] bits = 00b), and byte-sized transfer (MRA.SZ[1:0] bits = 00b). In the MRB register, specify incrementation of the destination address (MRB.DM[1:0] = 10b) and single data transfer by a single interrupt (MRB.CHNE bit = 0 and MRB.DISEL bit = 0). The MRB.DTS bit can be set to any value. Set the address of the RDR register in the SCI in the SAR register, the start address of the RAM area for data storage in the DAR register, and 128 (0080h) in the CRA register. The CRB register can be set to any value.

#### (2) DTC Vector Table Setting

The address where the transfer information for use with the RXI starts is set in the vector table for the DTC.

#### (3) ICU Set and DTC Module Activation

Set the corresponding ICU.DTCERn.DTCE bit to 1 and the ICU.IERi.IENj bit to 1.
Set the DTCST.DTCST bit to 1.

#### (4) SCI Setting

Enable the receive data full interrupt (RXI) by setting the SCR.RIE bit in the SCI to 1. If a reception error occurs during the SCI receive operation, further reception is not performed. Accordingly, make settings so that the CPU can accept receive error interrupts.

#### (5) DTC Transfer

Every time the reception of 1 byte by the SCI is completed, an RXI interrupt is generated to activate the DTC. The DTC transfers the received byte from the RDR of the SCI to RAM, after which the DAR register is incremented and the CRA register is decremented.

#### (6) Interrupt Handling

After 128 rounds of data transfer have been completed and the value in the CRA register becomes 0, an RXI interrupt request is generated for the CPU. Processing for completion is performed in the processing routine for this interrupt.

### 16.6.2 Chain Transfer when Counter = 0

The second data transfer is performed only when the counter = 0. Repeat transfer of a transfer count of 256 or more is enabled by the re-setting for the first data transfer.
The following shows an example of configuring a 128-Kbyte input buffer, where the input buffer is set so that its lower address starts with 0000h. Figure 16.15 shows a chain transfer when the counter = 0.

1. Set normal transfer mode for input data for the first data transfer. Set the following:
   Transfer source address: Fixed, the CRA register = 0000h (65,536 times), the MRB.CHNE bit = 1 (chain transfer is enabled), the MRB.CHNS bit = 1 (chain transfer is performed only when the transfer counter is 0), and the MRB.DISEL bit = 0 (an interrupt request to the CPU is generated when specified data transfer is completed).
2. Prepare the upper 8-bit address of the start address at every 65,536 times of the transfer destination address for the first data transfer in another area (such as ROM). For example, when setting the input buffer to 200000h to 21FFFFh, prepare 21h and 20h.
3. For the second data transfer, set repeat transfer mode (source side: repeat area) for re-setting the transfer destination address of the first data transfer. Specify the upper 8 bits of the DAR register in the first transfer information area for the transfer destination. At this time, set the MRB.CHNE bit = 0 (chain transfer is disabled) and the MRB.DISEL bit = 0 (an interrupt request to the CPU is generated when specified data transfer is completed). When setting the input buffer mentioned above to 200000h to 21FFFFh, set the transfer counter to 2.
4. The first data transfer is performed by an interrupt 65,536 times. When the transfer counter of the first data transfer becomes 0, the second data transfer starts. Set the upper 8 bits of the transfer source address of the first data transfer to 21h. The transfer counter (lower 16 bits) of the transfer destination address of the first data transfer is 0000h.
5. In succession, the first data transfer is performed by an interrupt 65,536 times specified for the first data transfer. When the transfer counter of the first data transfer becomes 0, the second data transfer starts. Set the upper 8 bits of the transfer source address of the first data transfer to 20h. The transfer counter (lower 16 bits) of the transfer destination address of the first data transfer is 0000h.
6. Steps 4 and 5 above are repeated infinitely. Since the second data transfer is in repeat transfer mode, no interrupt request to the CPU is generated.

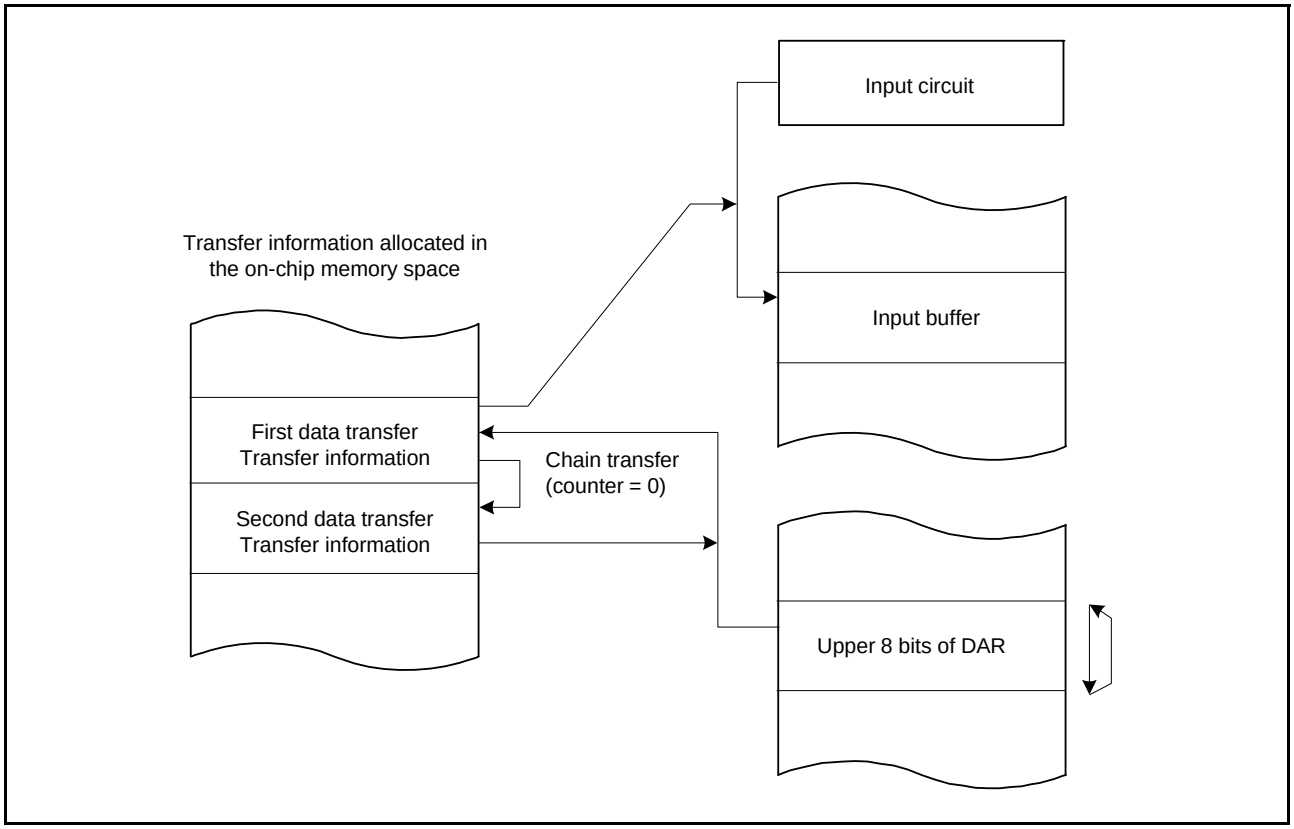



**Figure 16.15 Chain Transfer when Counter = 0**

## 16.7 Interrupt Source

When the DTC has finished data transfer of specified count or when data transfer with the MRB.DISEL set to 1 (an interrupt request to the CPU is generated each time DTC data transfer is performed) has been completed, an interrupt to the CPU is generated by the DTC startup source. Such interrupts to the CPU are controlled according to the PSW.I bit (interrupt enable) of the CPU, the PSW.IPL[3:0] bits (processor interrupt priority level), and the priority level of the interrupt controller.

## 16.8 Event Link

The DTC is capable of producing an event link request on completing transfer in response to one request.

## 16.9 Low Power Consumption Function

Before transition to the module stop state or software standby mode, clear the DTCST.DTCST bit to 0 (DTC module stop), and then perform the following.

### (1) Module Stop Function

Writing 1 to the MSTPCRA.MSTPA28 bit (transition to the module stop state is made) enables the module stop function of the DTC. If DTC transfer is in progress at the time 1 is written to the MSTPCRA.MSTPA28 bit, the transition to the module stop state proceeds after DTC transfer has ended. While the MSTPCRA.MSTPA28 bit is 1, accessing the DTC registers are prohibited.
Writing 0 to the MSTPCRA.MSTPA28 bit releases the DTC from the module stop state.

### (2) Software Standby Mode

Make settings in accord with the procedure under section 11.6.3.1, Entry to Software Standby Mode, in section 11, Low Power Consumption.
If DTC transfer operations are in progress at the time the WAIT instruction is executed, the transition to software standby mode follows the completion of DTC transfer.

### (3) Notes on Low Power Consumption Function

For the WAIT instruction and the register setting procedure, refer to section 11.7.5, Timing of WAIT Instructions in section 11, Low Power Consumption.
To perform DTC transfer after returning from a low power consumption mode, set the DTCST.DTCST to 1 again.
To use a request that is generated in software standby mode as an interrupt request to the CPU but not as a DTC startup request, specify the CPU as the interrupt request destination in accordance with the description in section 14.4.3, Selecting Interrupt Request Destinations in section 14, Interrupt Controller (ICUb), and then execute the WAIT instruction.

## 16.10 Usage Notes

### 16.10.1 Transfer Information Start Address

Be sure to set multiples of 4 for the transfer information start addresses in the vector table. Otherwise, such addresses are accessed with their lowest 2 bits regarded as 00b.

### 16.10.2 Allocating Transfer Information

Allocate transfer data in the memory area according to the endian of the area as shown in Figure 16.16.
For example, when writing CRA and CRB setting data with 16 bits in big endian, write the CRA setting data to lower address 0 and the CRB setting data to lower address 2. In little endian, write the CRB setting data to lower address 0 and the CRA setting data to lower address 2. When writing CRA and CRB setting data with 32 bits, place the CRA setting data at the MSB side and the CRB setting data at the LSB side regardless of endian, and then write the data to lower address 0.

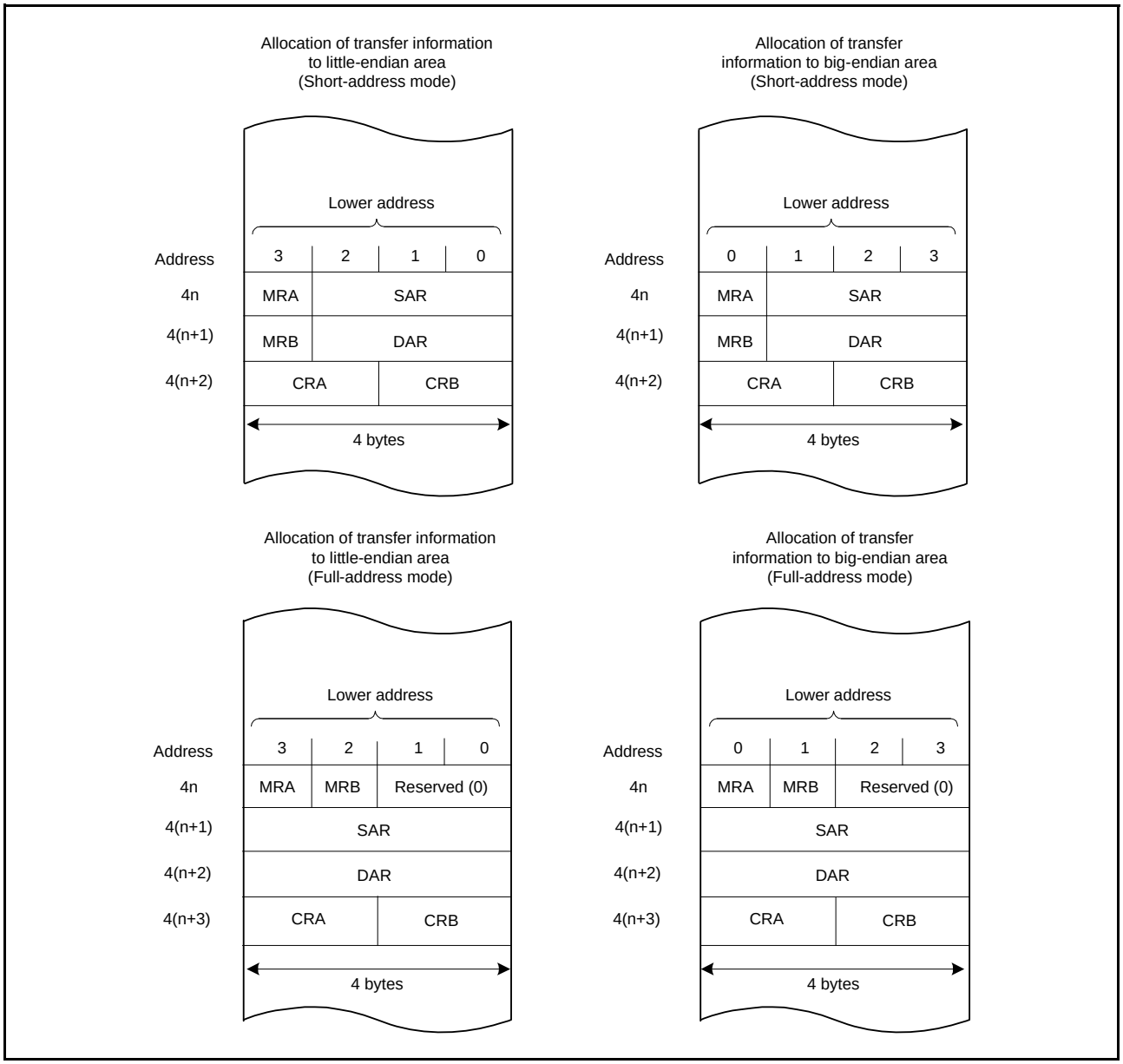



**Figure 16.16 Allocation of Transfer Information**

# 17. Event Link Controller (ELC)

## 17.1 Overview

The event link controller (ELC) connects (links) the events generated by various peripheral modules to different modules. Event linking allows direct cooperation between the modules without CPU intervention.

Table 17.1 lists the specifications of the ELC, and Figure 17.1 shows a block diagram of the ELC.

**Table 17.1 ELC Specifications**

| Item | Description |
|---|---|
| Event link function | • 36 types of event signals can be directly connected to modules.<br>• The operation of timer modules can be selected when an event is input to the timer module.<br>• Event link operation is possible for port B.<br>Single port*1: An event link can be set for a single bit specified in a port.<br>Port group*1: An event link can be set for a group of single bits specified within eight I/O ports. |
| Low power consumption function | Module stop state can be set. |

Note 1. The single port and port group specified as the input generate an event according to the change in the connected signal value. In products with 64-pin packages, when ports PC0 and PC1 are selected in port switching register A (PSRA), ports PB6 and PB7 of the event link controller (ELC) cannot be used as input and output events.
In products with 48-pin packages, when ports PC0 to PC3 are selected in port switching register B (PSRB), ports PB0, PB1, PB3, and PB5 of the event link controller (ELC) cannot be used as input and output events.

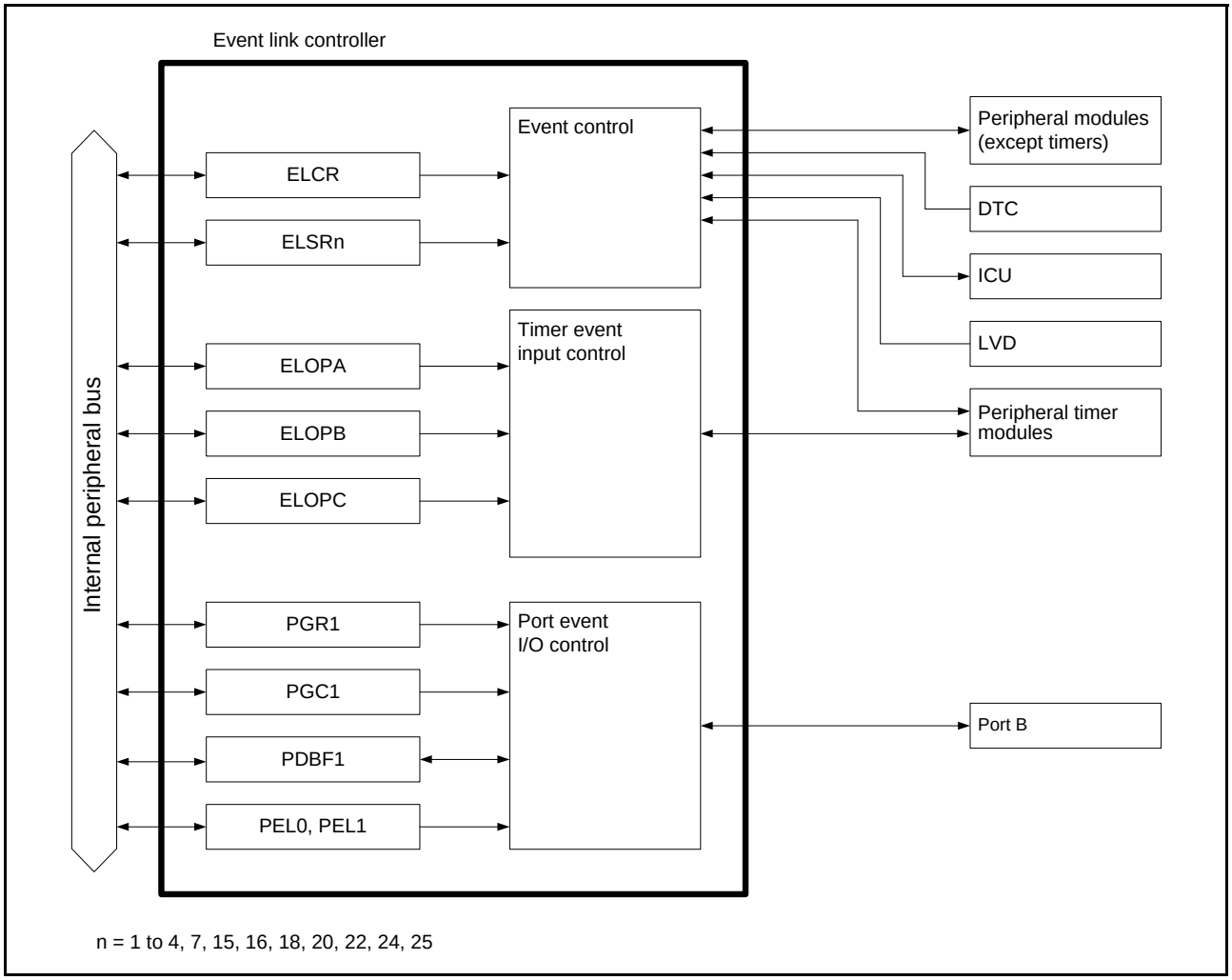



**Figure 17.1 ELC Block Diagram**

## 17.2 Register Descriptions

### 17.2.1 Event Link Control Register (ELCR)

Address(es): 0008 B100h

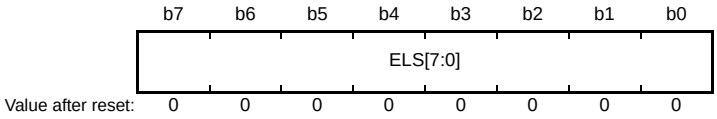
| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | ELCON | — | — | — | — | — | — | — |
| Value after reset: | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b6 to b0 | — | Reserved | These bits are read as 1. The write value should be 1. | R/W |
| b7 | ELCON | All Event Link Enable | 0: Linkage of all the events is disabled.<br>1: Linkage of all the events is enabled. | R/W |

The ELCR register controls operation of the event link controller (ELC).

## 17.2.2 Event Link Setting Register n (ELSRn) (n = 1 to 4, 7, 15, 16, 18, 20, 22, 24, 25)

Address(es): ELSR1: 0008 B102h, ELSR2: 0008 B103h, ELSR3: 0008 B104h, ELSR4: 0008 B105h, ELSR7: 0008 B108h, ELSR15: 0008 B110h, ELSR16: 0008 B111h, ELSR18: 0008 B113h, ELSR20: 0008 B115h, ELSR22: 0008 B117h, ELSR24: 0008 B119h, ELSR25: 0008 B11Ah




| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b7 to b0 | ELS[7:0] | Event Link Select | b7 b0<br>00000000: Event link function is stopped.<br>00001000 to 01101010: Set the number for the event signal to be linked.<br>Other than above: Setting prohibited | R/W |

The ELSRn register specifies an event signal to be linked to for each peripheral module. Table 17.2 shows the correspondence between the ELSRn register and the peripheral modules. Table 17.3 shows the correspondence between the event signal names set in the ELSRn register and the signal numbers.

**Table 17.2 Correspondence between the ELSRn Register and the Peripheral Functions**

| Register Name | Peripheral Function (Module) |
|---|---|
| ELSR1 | MTU1 |
| ELSR2 | MTU2 |
| ELSR3 | MTU3 |
| ELSR4 | MTU4 |
| ELSR7 | CMT1 |
| ELSR15 | 12-bit A/D converter |
| ELSR16 | DA0 |
| ELSR18 | Interrupt 1 |
| ELSR20 | Output port group 1 |
| ELSR22 | Input port group 1 |
| ELSR24 | Single port 0*1 |
| ELSR25 | Single port 1*1 |

Note 1. Do not set the DOC data operation condition met signal (ELS[7:0] bits = 6Ah) in the ELSR24 or ELSR25 register.

**Table 17.3 Correspondence between Event Signal Names Set in ELSRn.ELS[7:0] Bits and Signal Numbers**

| ELS[7:0] Bit Value | Name of Event Signal Set in ELSRn |
|---|---|
| 00001000 (08h) | MTU1 compare match 1A signal |
| 00001001 (09h) | MTU1 compare match 1B signal |
| 00001010 (0Ah) | MTU1 overflow signal |
| 00001011 (0Bh) | MTU1 underflow signal |
| 00001100 (0Ch) | MTU2 compare match 2A signal |
| 00001101 (0Dh) | MTU2 compare match 2B signal |
| 00001110 (0Eh) | MTU2 overflow signal |
| 00001111 (0Fh) | MTU2 underflow signal |
| 00010000 (10h) | MTU3 compare match 3A signal |
| 00010001 (11h) | MTU3 compare match 3B signal |
| 00010010 (12h) | MTU3 compare match 3C signal |
| 00010011 (13h) | MTU3 compare match 3D signal |
| 00010100 (14h) | MTU3 overflow signal |
| 00010101 (15h) | MTU4 compare match 4A signal |
| 00010110 (16h) | MTU4 compare match 4B signal |
| 00010111 (17h) | MTU4 compare match 4C signal |
| 00011000 (18h) | MTU4 compare match 4D signal |
| 00011001 (19h) | MTU4 overflow signal |
| 00011010 (1Ah) | MTU4 underflow signal |
| 00011111 (1Fh) | CMT1 compare match 1 signal |
| 00111010 (3Ah) | SCI5 error (receive error or error signal detection) signal |
| 00111011 (3Bh) | SCI5 receive data full signal |
| 00111100 (3Ch) | SCI5 transmit data empty signal |
| 00111101 (3Dh) | SCI5 transmit end signal |
| 01001110 (4Eh) | RIIC0 communication error or event generation signal |
| 01001111 (4Fh) | RIIC0 receive data full signal |
| 01010000 (50h) | RIIC0 transmit data empty signal |
| 01010001 (51h) | RIIC0 transmit end signal |
| 01011000 (58h) | A/D conversion end signal of 12-bit A/D converter |
| 01011011 (5Bh) | LVD1 voltage detection signal |
| 01100001 (61h) | DTC transfer end signal |
| 01100011 (63h) | Input edge detection signal of input port group 1 |
| 01100101 (65h) | Input edge detection signal of single input port 0 |
| 01100110 (66h) | Input edge detection signal of single input port 1 |
| 01101001 (69h) | Software event signal |
| 01101010 (6Ah) | DOC data operation condition met signal |
| Other than above: Setting prohibited | |

## 17.2.3 Event Link Option Setting Register A (ELOPA)

Address(es): 0008 B11Fh

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | MTU3MD[1:0] | | MTU2MD[1:0] | | MTU1MD[1:0] | | — | — |
| Value after reset: | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b1, b0 | — | Reserved | These bits are read as 1. The write value should be 1. | R/W |
| b3, b2 | MTU1MD[1:0] | MTU1 Operation Select | b3 b2<br>0 0: Counting is started.<br>0 1: Counting is restarted.<br>1 0: Input capture*1<br>1 1: Event is disabled. | R/W |
| b5, b4 | MTU2MD[1:0] | MTU2 Operation Select | b5 b4<br>0 0: Counting is started.<br>0 1: Counting is restarted.<br>1 0: Input capture*2<br>1 1: Event is disabled. | R/W |
| b7, b6 | MTU3MD[1:0] | MTU3 Operation Select | b7 b6<br>0 0: Counting is started.<br>0 1: Counting is restarted.<br>1 0: Input capture*3<br>1 1: Event is disabled. | R/W |

Note 1. The MTU1.TCNT value is captured into MTU1.TGRA.
Note 2. The MTU2.TCNT value is captured into MTU2.TGRA.
Note 3. The MTU3.TCNT value is captured into MTU3.TGRA.

ELOPA determines the operation of MTU1 to MTU3 in the MTU when an event is input. All events must be disabled when the ELC function is not to be used.

## 17.2.4 Event Link Option Setting Register B (ELOPB)

Address(es): 0008 B120h

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | MTU4MD[1:0] | |
| Value after reset: | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b1, b0 | MTU4MD[1:0] | MTU4 Operation Select | b1 b0<br>0 0: Counting is started.<br>0 1: Counting is restarted.<br>1 0: Input capture*1<br>1 1: Event is disabled. | R/W |
| b7 to b2 | — | Reserved | These bits are read as 1. The write value should be 1. | R/W |

Note 1. The MTU4.TCNT value is captured into MTU4.TGRA.

ELOPB determines the operation of MTU4 in the MTU when an event is input. All events must be disabled when the ELC function is not to be used.

## 17.2.5 Event Link Option Setting Register C (ELOPC)

Address(es): 0008 B121h

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | CMT1MD[1:0] | | — | — |
| Value after reset: | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b1, b0 | — | Reserved | These bits are read as 1. The write value should be 1. | R/W |
| b3, b2 | CMT1MD[1:0] | CMT1 Operation Select | b3 b2<br>0 0: Counting is started.<br>0 1: Counting is restarted.<br>1 0: Event counter<br>1 1: Event is disabled. | R/W |
| b7 to b4 | — | Reserved | These bits are read as 1. The write value should be 1. | R/W |

ELOPC determines the operation of CMT1 in the CMT when an event is input. All events must be disabled when the ELC function is not to be used.

## 17.2.6 Port Group Setting Register 1 (PGR1)

Address(es): PGR1: 0008 B123h

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | PGR7 | PGR6 | PGR5 | PGR4 | PGR3 | PGR2 | PGR1 | PGR0 |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b0 | PGR0 | Port Group Setting 0 | 0: The port bit is not specified as a member of the same group.<br>1: The port bit is specified as a member of the same group. | R/W |
| b1 | PGR1 | Port Group Setting 1 | | R/W |
| b2 | PGR2 | Port Group Setting 2 | | R/W |
| b3 | PGR3 | Port Group Setting 3 | | R/W |
| b4 | PGR4 | Port Group Setting 4 | | R/W |
| b5 | PGR5 | Port Group Setting 5 | | R/W |
| b6 | PGR6 | Port Group Setting 6 | | R/W |
| b7 | PGR7 | Port Group Setting 7 | | R/W |

PGR1 specifies a group for I/O port bits. PGR1 specifies each port bit in the same eight I/O ports as the member of a group. One to eight port bits can be specified as the members of the same group as required. The correspondence between PGR1 and ports is shown in Table 17.4.

## 17.2.7 Port Group Control Register 1 (PGC1)

Address(es): PGC1: 0008 B125h

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | PGCO[2:0] | | | — | PGCOVE | PGCI[1:0] | |
| Value after reset: | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b1, b0 | PGCI[1:0] | Event Output Edge Select | b1 b0<br>0 0: Event is generated upon detection of the rising edge of the external input signal.<br>0 1: Event is generated upon detection of the falling edge of the external input signal.<br>1 X: Event is generated upon detection of both the rising and falling edges of the external input signal. | R/W |
| b2 | PGCOVE | PDBF Overwrite | 0: Overwriting PDBF register is disabled.<br>1: Overwriting PDBF register is enabled. | R/W |
| b3 | — | Reserved | This bit is read as 1. The write value should be 1. | R/W |
| b6 to b4 | PGCO[2:0] | Port Group Operation Select | b6 b4<br>0 0 0: 0 is output when the event is input.<br>0 0 1: 1 is output when the event is input.<br>0 1 0: The toggled (inverted) value is output when the event is input.<br>0 1 1: The buffer value is output when the event is input.<br>1 X X: The bit value is rotated out in the group (from MSB to LSB) when the event is input. | R/W |
| b7 | — | Reserved | This bit is read as 1. The write value should be 1. | R/W |

X: Don't care

For the output port group, PGC1 specifies the form of outputting the signal externally via the port when the event signal is input. For the input port group, PGC1 enables/disables overwriting of PDBF and specifies the conditions of event generation (edge of the externally input signal).
The correspondence between PGC1 and ports is shown in Table 17.4.

## 17.2.8 Port Buffer Register 1 (PDBF1)

Address(es): PDBF1: 0008 B127h

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | PDBF7 | PDBF6 | PDBF5 | PDBF4 | PDBF3 | PDBF2 | PDBF1 | PDBF0 |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b0 | PDBF0 | Port Buffer 0 | Data is transferred between PODR and PDBF when an event is input. Write access to the bit specified as a member of the input port group is invalid. For details, refer to section 17.3, Operation. | R/W |
| b1 | PDBF1 | Port Buffer 1 | | R/W |
| b2 | PDBF2 | Port Buffer 2 | | R/W |
| b3 | PDBF3 | Port Buffer 3 | | R/W |
| b4 | PDBF4 | Port Buffer 4 | | R/W |
| b5 | PDBF5 | Port Buffer 5 | | R/W |
| b6 | PDBF6 | Port Buffer 6 | | R/W |
| b7 | PDBF7 | Port Buffer 7 | | R/W |

PDBF1 is an 8-bit readable/writable register used in combination with PGR1. For PDBF1 operations, refer to section 17.3, Operation. Table 17.4 shows register related to port groups and corresponding port numbers.

**Table 17.4 Register Related to Port Groups and Corresponding Port Numbers**

| Port Group Setting Register (PGR) | Port Group Control Register (PGC) | Port Buffer Register (PDBF) | Port Number |
|---|---|---|---|
| PGR1 register | PGC1 register | PDBF1 register | Port B |

## 17.2.9 Event Link Port Setting Register n (PELn) (n = 0, 1)

Address(es): PEL0: 0008 B129h, PEL1: 0008 B12Ah

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | PSM[1:0] | | PSP[1:0] | | PSB[2:0] | | |
| Value after reset: | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b2 to b0 | PSB[2:0] | Bit Number Specification | A bit number in eight I/O ports is specified. | R/W |
| b4, b3 | PSP[1:0] | Port Number Specification | b4 b3<br>0 0: Setting is invalid.<br>0 1: Port B (corresponding to PGR1)<br>1 X: Setting prohibited | R/W |
| b6, b5 | PSM[1:0] | Event Link Specification | • For the output port, data to be output from the port is specified.<br>b6 b5<br>0 0: 0 is output when the event is input.<br>0 1: 1 is output when the event is input.<br>1 X: The toggled (inverted) value is output when the event is input.<br>• For the input port, the edge on which the event is to be output is specified.<br>b6 b5<br>0 0: Event is output upon detection of the rising edge.<br>0 1: Event is output upon detection of the falling edge.<br>1 X: Event is output upon detection of both the rising and falling edges. | R/W |
| b7 | — | Reserved | This bit is read as 1. The write value should be 1. | R/W |

X: Don't care

PELn specifies the 1-bit port (hereinafter referred to as a single port) to which an event is to be linked, the port operation upon the event signal input, and the conditions of event generation. In the RX111 Group, a total of 2 bits in port B can be specified as single ports.

## 17.2.10 Event Link Software Event Generation Register (ELSEGR)

Address(es): 0008 B12Dh

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | WI | WE | — | — | — | — | — | SEG |
| Value after reset: | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b0 | SEG | Software Event Generation | 0: Normal operation<br>1: Software event is generated. | W |
| b5 to b1 | — | Reserved | These bits are read as 1. The write value should be 1. | R/W |
| b6 | WE | SEG Bit Write Enable | 0: Write to SEG bit is disabled.<br>1: Write to SEG bit is enabled. | R/W |
| b7 | WI | ELSEGR Register Write Disable | 0: Write to ELSEGR register is enabled.<br>1: Write to ELSEGR register is disabled. | W |

The MOV instruction must always be used to write to this register.

### SEG Bit (Software Event Generation)

When 1 is written to this bit while the WE bit is 1, a software event is generated.
This bit is read as 0. Even if 1 is written to this bit, the data will not be stored.

### WE Bit (SEG Bit Write Enable)

The SEG bit can be written to only when the WE bit is 1.
[Setting condition]
If 1 is written to this bit while the WI bit is 0, this bit becomes 1.
[Clearing condition]
If 0 is written to this bit while the WI bit is 0, this bit becomes 0.

### WI Bit (ELSEGR Register Write Disable)

The ELSEGR register can be written to only when the value to be written to the WI bit is 0.
This bit is read as 1.

## 17.3 Operation

### 17.3.1 Relation between Interrupt Handling and Event Linking

The modules incorporated in the RX111 are provided with the interrupt request status flags and the bits to enable/disable these interrupt requests. When an interrupt request is generated in a module, the corresponding interrupt request status flag is set. If the corresponding interrupt request is enabled then, the interrupt requested is issued to the CPU.
In contrast, the ELC uses interrupt requests (hereinafter referred to as events) generated in modules as event signals that directly activate other modules. This means that the event signal can be used whether or not the interrupt signal is enabled. Figure 17.2 shows the relation between the interrupt handling and ELC.

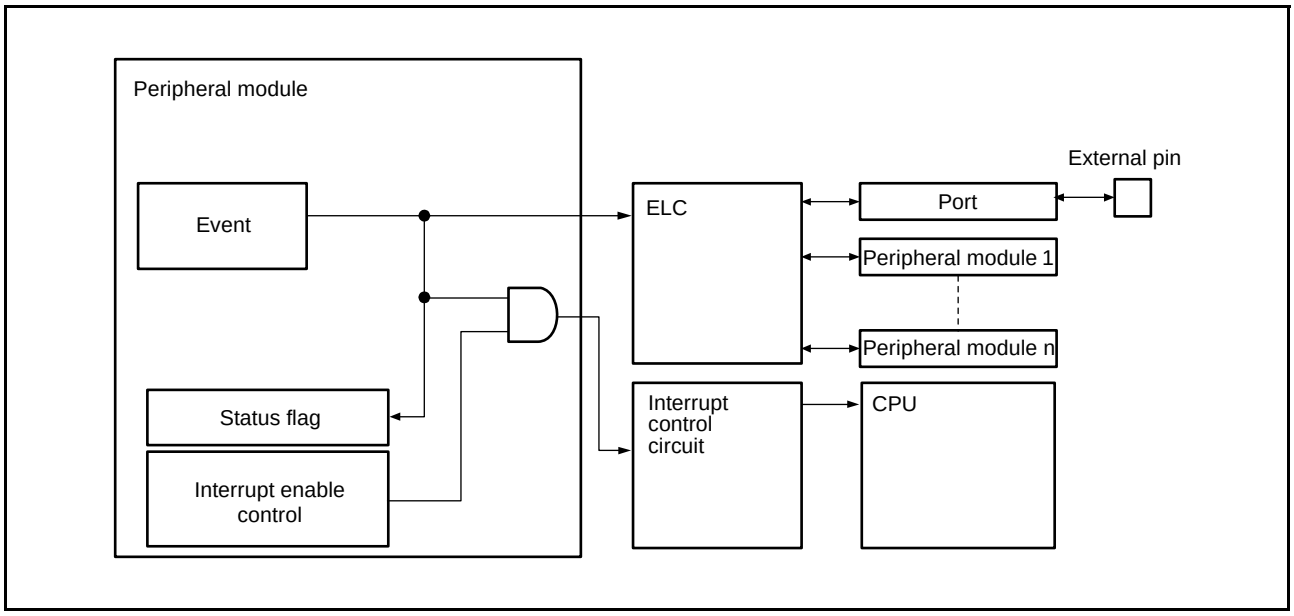



**Figure 17.2 Relation between Interrupt Handling and ELC**

### 17.3.2 Event Linkage

When an event has been set as a trigger in an event link setting register (ELSRn) and then occurs, that event is linked with the corresponding module (the module is activated). Only one type of event can be connected with one module. When a module is to be activated by the ELC, the operation of the module must be set up in advance. Table 17.5 lists the operations of modules when an event is input.

**Table 17.5 Operations of Modules When Event is Input**

| Module | Operations when Event is Input | | |
|---|---|---|---|
| MTU<br>CMT | Each timer operates differently depending on the ELOPA to ELOPC registers as below.<br>• Starts counting when an event signal is input.<br>• Restarts counting when an event signal is input.<br>• Counts the input events (CMT).<br>• Performs input-capture operation when an event is input (MTU). | | |
| A/D converter | Starts A/D conversion when an event signal is input. | | |
| D/A converter | Starts D/A conversion when an event signal is input | | |
| I/O ports (output) | The value of PODR (port output data register) changes when an event signal is input. (The value output from the relevant external pin changes.) | Port group | The port group operates differently depending on the settings as below.<br>• Changes the PODR value to the specified value.<br>• Transfers the PDBF1 value to the PODR register.<br>• Rotates out the bit value. |
| | | Single port | Changes the PODR value to the specified value. |
| I/O ports (input) | When the signal value of the input pin changes | Port group | Generates an event. |
| | | Single port | |
| | When an event is input | Port group | Transfers the signal value of the external pin to the PDBF1 register. |
| | | Single port | Event connection is not possible. |
| Interrupt controller | Issues an event to the CPU and starts DTC data transfer. | | |

### 17.3.3 Operation of Peripheral Timer Modules When Event is Input

The operations are performed depending on the ELOPA to ELOPC registers when an event is input.

#### (1) Counting Start Operation

When an event is input, the timer starts counting, which sets the count start bit*1 in each timer control register to 1. An event that is input while the count start bit is 1 is invalid.

#### (2) Counting Restart Operation

When an event is input, the timer counter*1 is initialized. Since the count start bit*1 in each timer control register is retained, counting is restarted when an event is input while the count start bit is set to 1.

#### (3) Event Counter Operation

Event input is selected as the timer clock source and the timer counts events.

#### (4) Input Capture Operation

When an event is input, the timer performs input-capture operation.

Note 1. Refer to the descriptions on the bit in the relevant timer section.

### 17.3.4 Operation of A/D and D/A Converters when Event is Input

The A/D and D/A converter start A/D and D/A conversion, respectively, when the ADCSR.ADST bit and the DACR.DAOE0 bit*1 are set to 1.

Note 1. Refer to the descriptions on the bits in the A/D and D/A converter sections.

### 17.3.5 I/O Port Operation upon Event Input and Event Generation

The I/O port operation to be performed upon event input to the port can be set and the operation causing the port to generate an event can be set.

#### (1) Single ports and Port Groups

There are two event link modes: event link to single ports and event link to port groups. In the former mode, events can be connected to eight I/O ports. In the latter mode, events can be connected to port groups consisting of any two or more bits in the same eight I/O ports.
A single port can be set by specifying any bit in the I/O port*1 to which an event can be connected using the PEL0 and PEL1 registers. A port group can be set by specifying any 2 or more bits in the I/O port*1 to which an event can be connected using the PGC1 register. One input port group and one output port group can be set in the same I/O port.
If the I/O port bit is specified as both a single port and a member of a port group, both functions are enabled when the relevant port is input, whereas only the port group function is enabled when the relevant port is output.
Set the PDR register to select the direction of the I/O ports.

Note 1. Port B

### (2) Event Generation by Single Input Ports

An single input port generates an event when the signal value of the external pin connected to the relevant port changes. The event generation condition is specified using the PEL0 and PEL1 registers. An example of operation is shown in Figure 17.3.

### (3) Single Output Ports Operation upon Event Input

When an event is input to a single output port, the PODR value of the relevant port changes. The specific change of the PODR value is specified using the PEL0 and PEL1 registers. Thus, the change of the PODR value changes the signal value of the external pin connected to the relevant port. An example of operation is shown in Figure 17.3.

### (4) Input Port Group Operation upon Event Input and Event Generation

An input port group generates an event when the signal value of any one of the external pins connected to the relevant port group changes. The event generation condition is specified using the PGC1 register. When an event is input to an input port group, the signal value of the external pin upon event input is transferred to the PDBF1 register. In this case, only the values of the bits specified as members of the input port group are transferred. An example of operation is shown in Figure 17.4.

### (5) Output Port Group Operation upon Event Input

When an event is input to an output port group, the PODR values change to the values according to the PGC1 settings. An example of operation is shown in Figure 17.5.

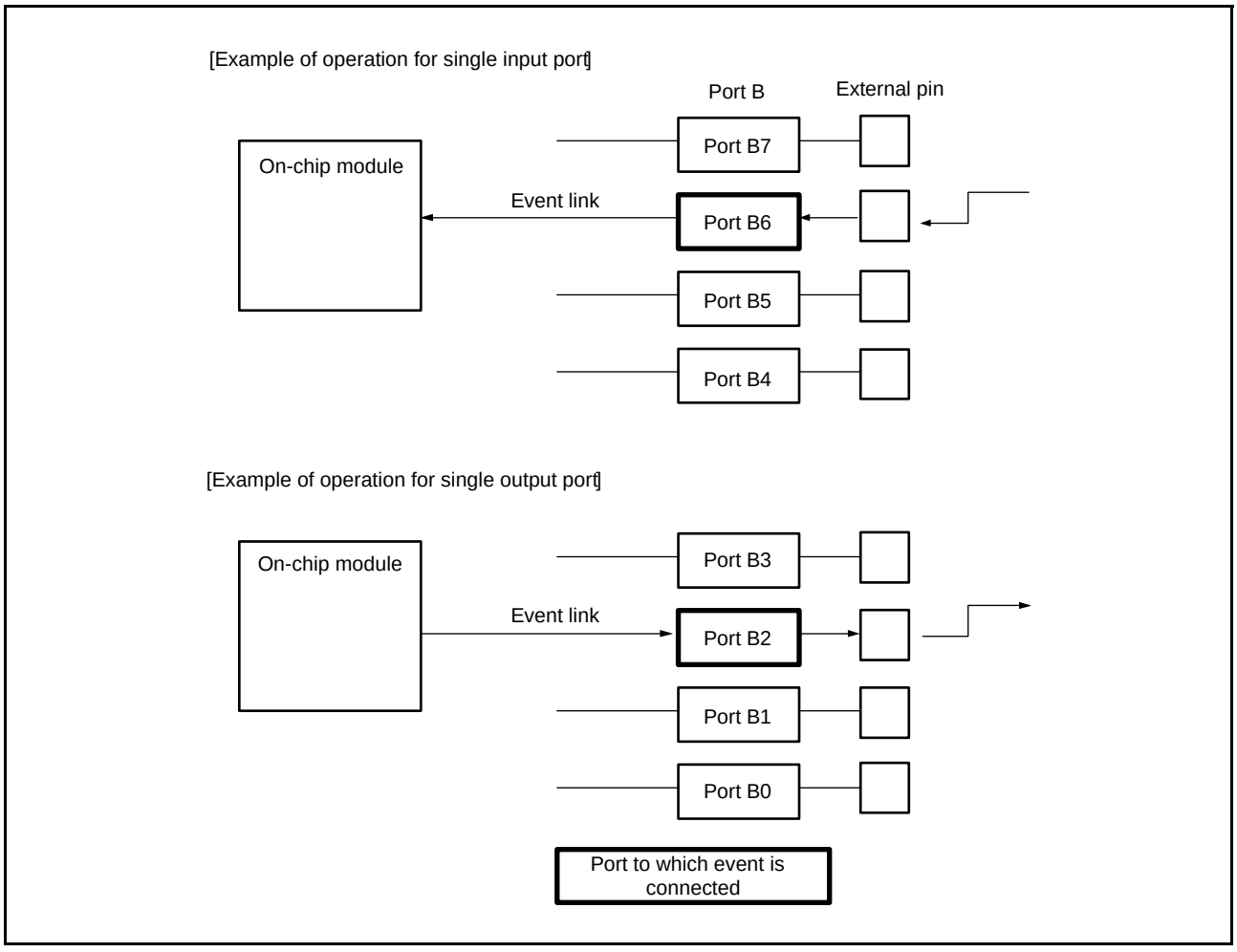



**Figure 17.3 Event Linkage Related to Single Ports**

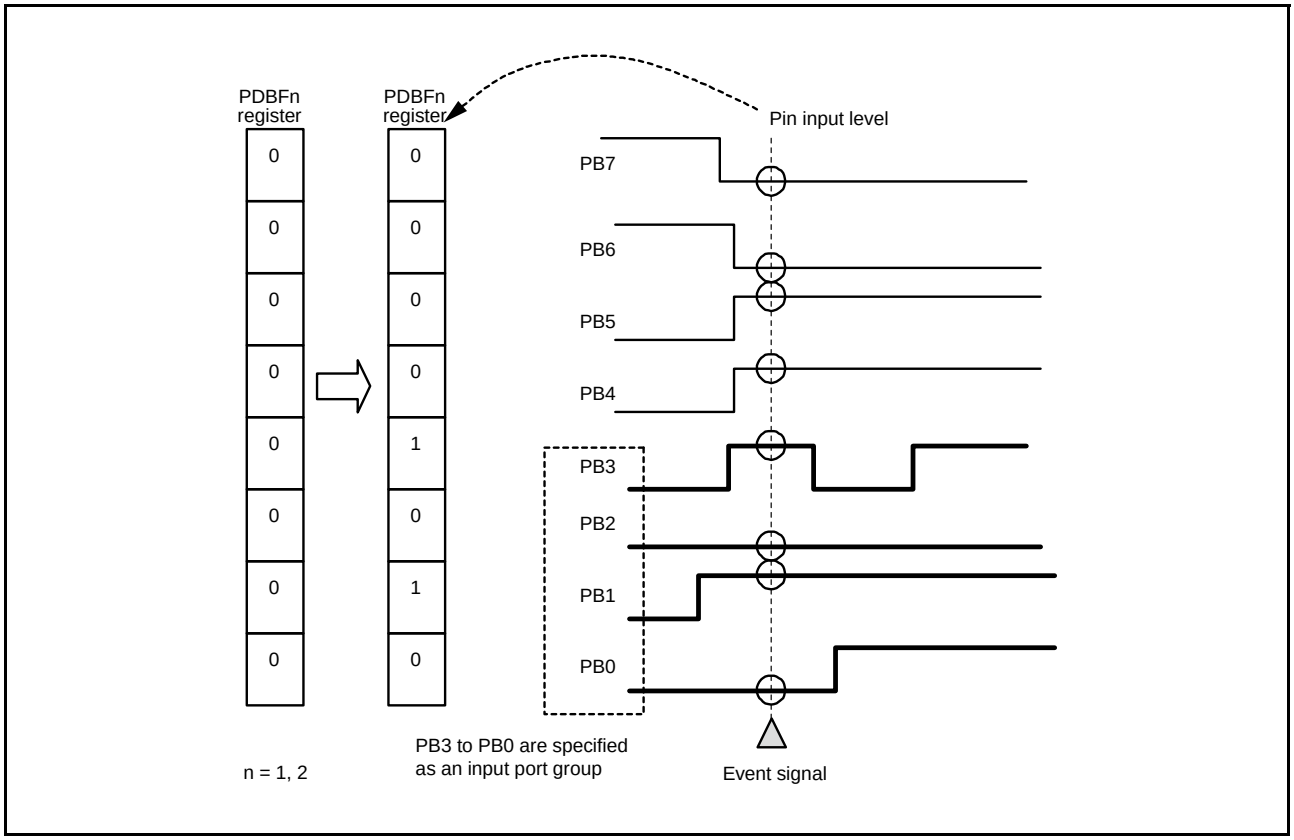



**Figure 17.4 Event Linkage Related to Input Port Groups**

## (6) Operation of Port Buffer Registers

### (a) Input Port Groups

When an event is input to an input port group, the signal value of the external pin of the bit specified as a member of the input port group is transferred to the PDBF1 register. If another event is input to the input port group in this state, operations are performed depending on the PGC1.PGCOVE bit setting as described below.

- PGC1.PGCOVE = 0 (overwriting is disabled)
  If the PDBF1 value that has been transferred upon the latest event input has already been read by the CPU (or transferred by the DTC), the signal value of the external pin is transferred to the PDBF1 register. If not read, the signal value of the external pin is not transferred and the input event is invalid.
- PGC1.PGCOVE = 1 (overwriting is enabled)
  When another event is input to an input port group, the signal value of the external pin is transferred to the PDBF1 register.

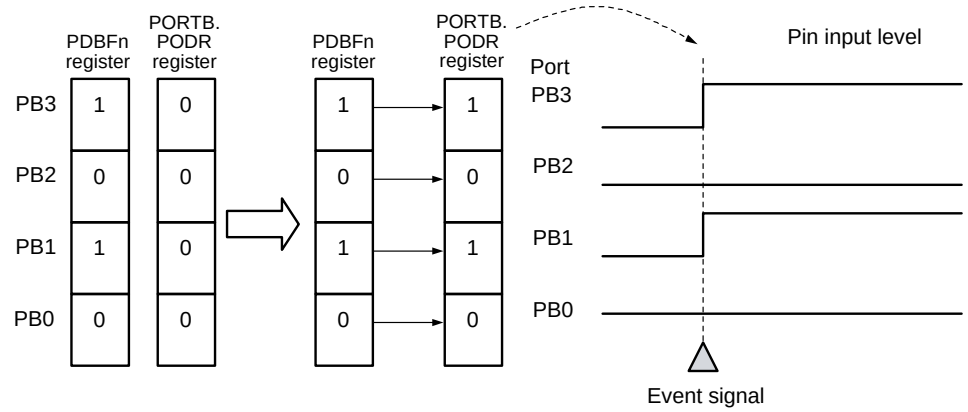
### (b) Output Port Groups

If an output port group is specified so that it should output the PDBF1 value, the PDBF1 value is transferred to the PODR register when an event is input to the output port group. In this case, only the values of the bits specified as members of the output port group are transferred.
If an output port group is specified so that it should rotate out the bit values in the group (PGC1.PGCO[2:0] bits = 1XX), the PDBF1 data is transferred to the PODR register, and then the PODR value is rotated bit by bit from MSB to LSB. The initial value to be output to the port group should be provided in the PDBF1 register.
Examples of operation are shown in Figure 17.5 and Figure 17.6.

### (7) Restrictions on Writing to PODR and PDBF Registers by CPU

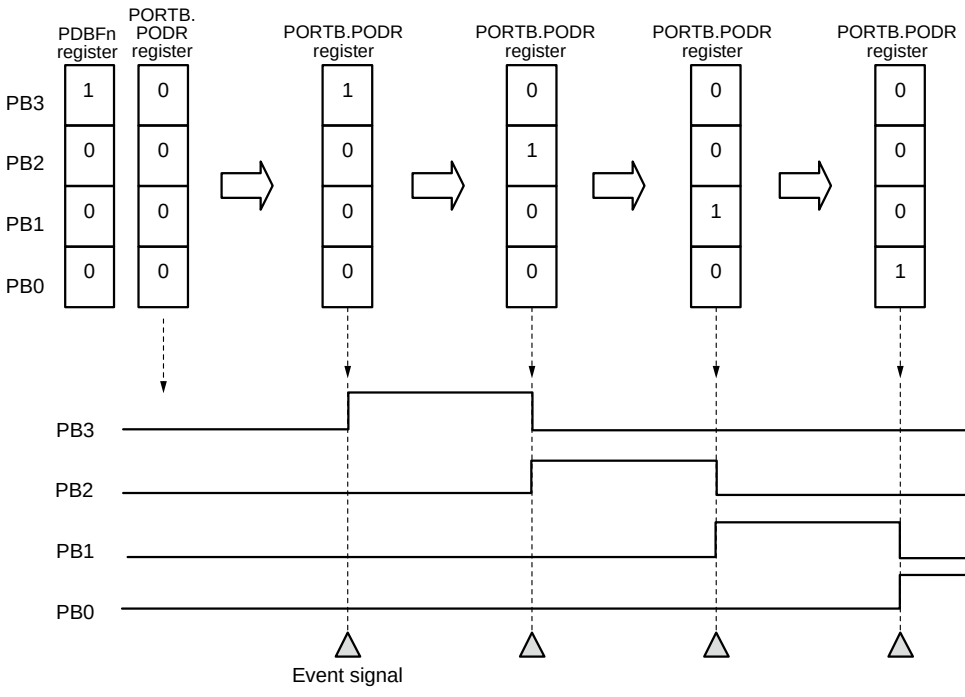
When the ELCR.ELCON bit is set to 1, write access to the following registers is invalid.

- If bits are specified as members of the input port group and the event linkage is set for the port group, write access to the relevant bits in the PDBF1 register is invalid. However, when the DOC is selected for event input, write access is not invalid.
- If port bits are specified as members of the output port group, write access to the relevant bits in the PODR register is invalid.
- If a port bit is specified as a single output port and the event linkage is set (by the ELSRn register) for the port, write access to the relevant bit in the PODR register is invalid. However, when the DOC is selected for event input, write access is not invalid.




n = 1, 2
Note: • PB3 to PB0 are specified as an output port group.

**Figure 17.5 Event Linkage Related to Output Port Groups**




n = 1, 2
Note: • PB3 to PB0 are specified as an output port group.

**Figure 17.6 Bit-Rotating Operation of Output Port Groups**

### 17.3.6 Procedure for Linking Events

The following describes the procedure for linking events.

1. Set the operation of the module to which an event is to be linked.
2. If events are linked to ports, set the registers corresponding to the ports as below.
   PODR: Set the initial values of the output ports.
   PDR: Set the I/O direction of the ports.
   PGR1: If ports are used as a port group, set the ports (in bit units) to be grouped.
   PGC1: Set the operation of the port group.
   PEL0, PEL1: If ports are used as single ports, set the ports, the operation of the ports when an event is input, and the condition when an event is generated.
3. To the ELSRn register corresponding to the module to which an event signal is to be linked, set the number of the event signal.
4. If events are to be linked to timer modules, set the ELOPA to ELOPC registers corresponding to the timers as required.
5. Set the ELCR.ELCON bit to 1, which enables linkage of all the events.
6. Set the operation of the module from which an event is output, and start the module. This allows the event output from the module to start the module to which an event is linked as specified.
7. To stop event linkage of some independent modules, set 00000000b to the ELSRn.ELS[7:0] bits corresponding to the modules. To stop linkage of all the events, clear the ELCR.ELCON bit to 0.

## 17.4 Usage Notes

### 17.4.1 Setting ELSRn Register

#### (1) Setting ELSR18 Register

Specify an event number from among 01100011 (63h) to 01101010 (6Ah); the other settings are prohibited.

#### (2) Setting ELSR24 and ELSR25 Registers

Setting the DOC data operation condition met signal (01101010 (6Ah)) is prohibited.

### 17.4.2 Setting Bit-Rotating Operation of Output Port Groups

When the values of the PDBF1 register are changed in the bit-rotating operation mode of the output port group, set the ELSRn register again. When events are used during bit-rotating operation, generate an event after an interval of one PCLK cycle. If not, the normal operation cannot be provided.

### 17.4.3 Linking DTC Transfer End Signals as Events

When linking the DTC transfer end signals as events, do not set the same peripheral module as the DTC transfer destination and event link destination. If set, the peripheral module might be started before DTC transfer to the peripheral module is completed.

### 17.4.4 Setting Clocks

To connect events output by the peripheral modules with each other and operate the modules, it is necessary for the ELC and the related modules to be enabled. The modules cannot operate if the related modules are in the module stop state or in the specific low power consumption mode in which the module is stopped (software standby mode).

### 17.4.5 Enabling and Disabling the ELC

Set the MSTPCRB.MSTPB9 bit to enable and disable ELC operation. ELC operation is disabled after a reset. ELC registers can be accessed by setting the MSTPCRB.MSTPB9 bit to 0 (ELC module stop state is canceled).

# 18. I/O Ports

## 18.1 Overview

The I/O ports can function as general I/O ports, I/O pins of a peripheral module, or as input pins for an interrupt.
Each pin can also be configured as an I/O pin of a peripheral module or as an input pin for an interrupt. Immediately after a reset, all pins function as input pins, and pin functions are switched by register settings. The setting of each pin is specified by the registers for the corresponding I/O port and registers for the on-chip peripheral modules.
Each port has the port direction register (PDR) that selects input or output direction, the port output data register (PODR) that holds data for output, the port input data register (PIDR) that indicates the pin states, the open drain control register y (ODRy, y = 0, 1) that selects the output type of each pin, the pull-up control register (PCR) that controls on/off of the input pull-up MOS, and the port mode register (PMR) that specifies the pin function of each port.
For details on the PMR register, refer to section 19, Multi-Function Pin Controller (MPC).
Products with 64-pin packages include port switching register A (PSRA) and products with 48-pin packages include port switching register B (PSRB). However, when ports PC0 and PC1 are selected in port switching register A (PSRA), input to ports PB6 and PB7 and the output port event function of the ELC cannot be used. When ports PC0 to PC3 are selected in port switching register B (PSRB), input to ports PB0, PB1, PB3, and PB5 and the output port event function of the ELC cannot be used.
The configuration of the I/O ports differs depending on the package. Table 18.1 lists the specifications of I/O ports, and Table 18.2 lists the port functions.

**Table 18.1 I/O Port Specifications**

| Port Symbol | Package | | Package | | Package | | Package | |
|---|---|---|---|---|---|---|---|---|
| | 64 Pins | Number of Pins | 48 Pins | Number of Pins | 40 Pins | Number of Pins | 36 Pins | Number of Pins |
| PORT0 | P03, P05 | 2 | Not available | 0 | Not available | 0 | Not available | 0 |
| PORT1 | P14, P15, P16, P17 | 4 | P14, P15, P16, P17 | 4 | P14, P15, P16, P17 | 4 | P14, P15, P16, P17 | 4 |
| PORT2 | P26, P27 | 2 | P26, P27 | 2 | P26, P27 | 2 | P27 | 1 |
| PORT3 | P30 to P32, P35 | 4 | P35 | 1 | P32, P35 | 2 | P35 | 1 |
| PORT4 | P40 to P44, P46 | 6 | P40 to P42, P46 | 4 | P41, P42, P46 | 3 | P41, P42 | 2 |
| PORT5 | P54, P55 | 2 | Not available | 0 | Not available | 0 | Not available | 0 |
| PORTA | PA0, PA1, PA3, PA4, PA6 | 5 | PA1, PA3, PA4, PA6 | 4 | PA1, PA3, PA4, PA6 | 4 | PA3, PA4, PA6 | 3 |
| PORTB | PB0, PB1, PB3, PB5 to PB7 | 6 | PB0, PB1, PB3, PB5 | 4 | PB0, PB3 | 2 | PB0, PB3 | 2 |
| PORTC | PC0 to PC7 | 8 | PC0 to PC7 | 8 | PC4 | 1 | PC4 | 1 |
| PORTE | PE0 to PE7 | 8 | PE0 to PE4, PE7 | 6 | PE0 to PE4 | 5 | PE0 to PE4 | 5 |
| PORTH | PH7 | 1 | PH7 | 1 | Not available | 0 | Not available | 0 |
| PORTJ | PJ6, PJ7 | 2 | PJ6, PJ7 | 2 | PJ6, PJ7 | 2 | PJ6, PJ7 | 2 |
| | Total number of pins | 50 | Total number of pins | 36 | Total number of pins | 25 | Total number of pins | 21 |

**Table 18.2 Port Functions**

| Port | Port Register | Input Pull-up | Open Drain Output | 5-V Tolerant |
|---|---|---|---|---|
| PORT0 | P03, P05 | ○ | — | — |
| PORT1 | P14, P15 | ○ | ○ | — |
| | P16, P17 | ○ | ○ | ○ |
| PORT2 | P26, P27 | ○ | ○ | — |
| PORT3 | P30 to P32 | ○ | ○ | — |
| | P35 | — | — | — |
| PORT4 | P40 to P44, P46 | — | — | — |
| PORT5 | P54, P55 | ○ | — | — |
| PORTA | PA0, PA1, PA3, PA4 | ○ | ○ | — |
| | PA6 | ○ | ○ | ○ |
| PORTB | PB1, PB3, PB5 to PB7 | ○ | ○ | — |
| | PB0 | ○ | ○ | ○ |
| PORTC | PC0 to PC7 | ○ | ○ | — |
| PORTE | PE0 to PE7 | ○ | ○ | — |
| PORTH | PH7 | — | — | — |
| PORTJ | PJ6, PJ7 | — | — | — |

Specifying input pull-up, open-drain output, or 5-V tolerance is available for other signals on pins that also function as general I/O pins.

## 18.2 I/O Port Configuration

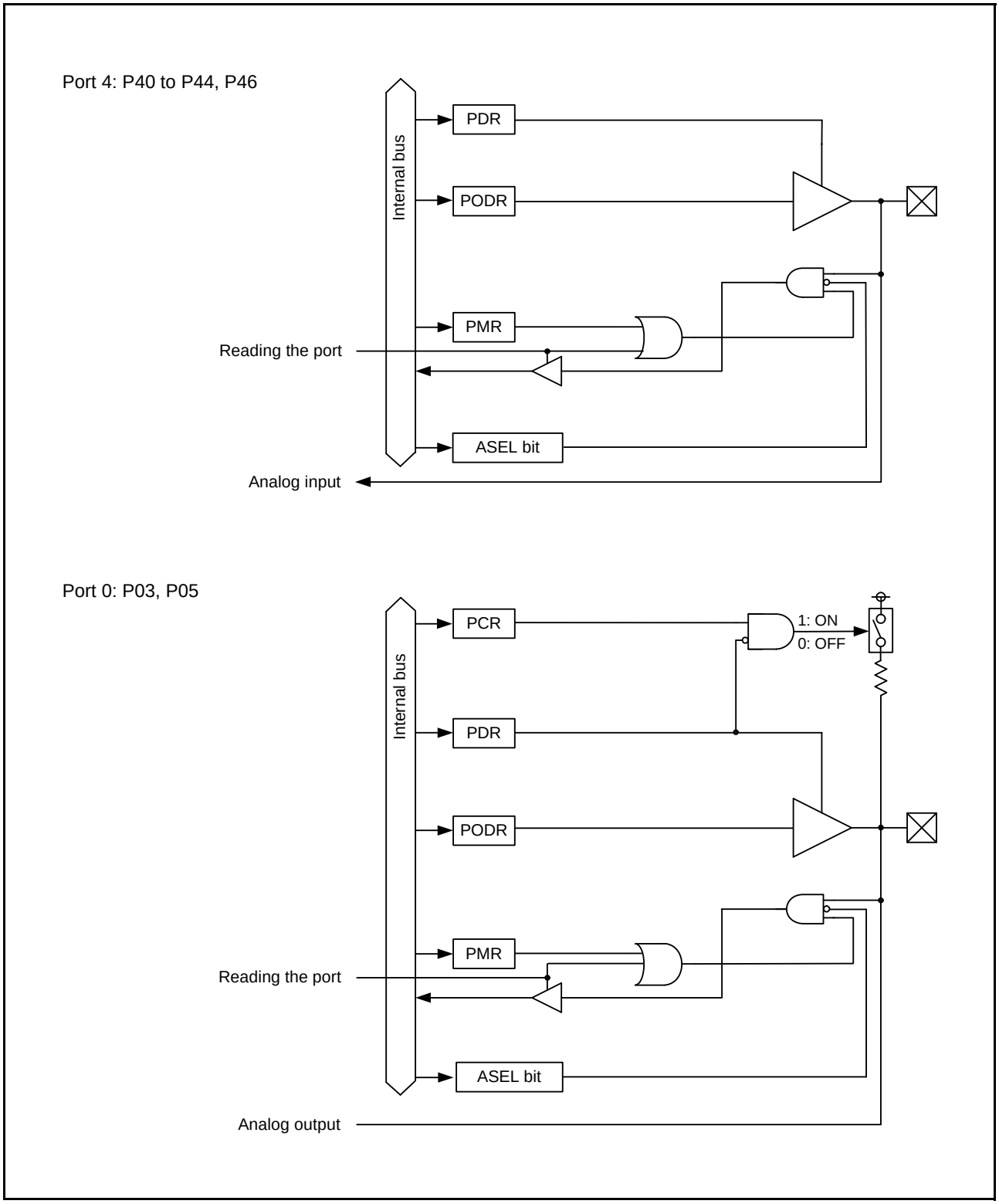
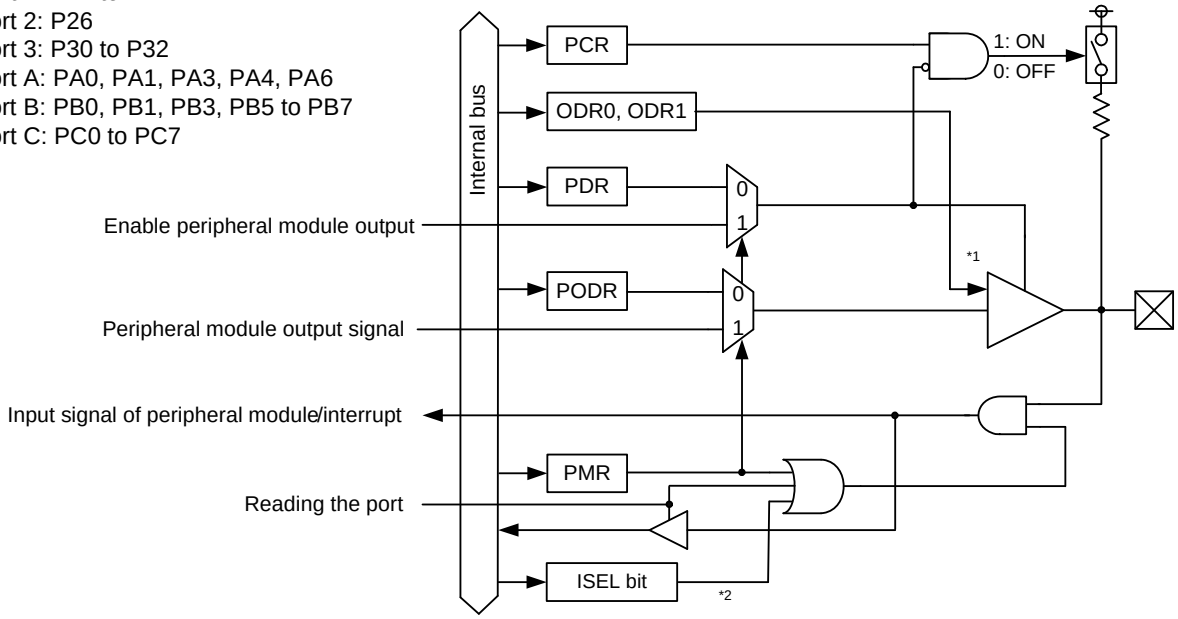


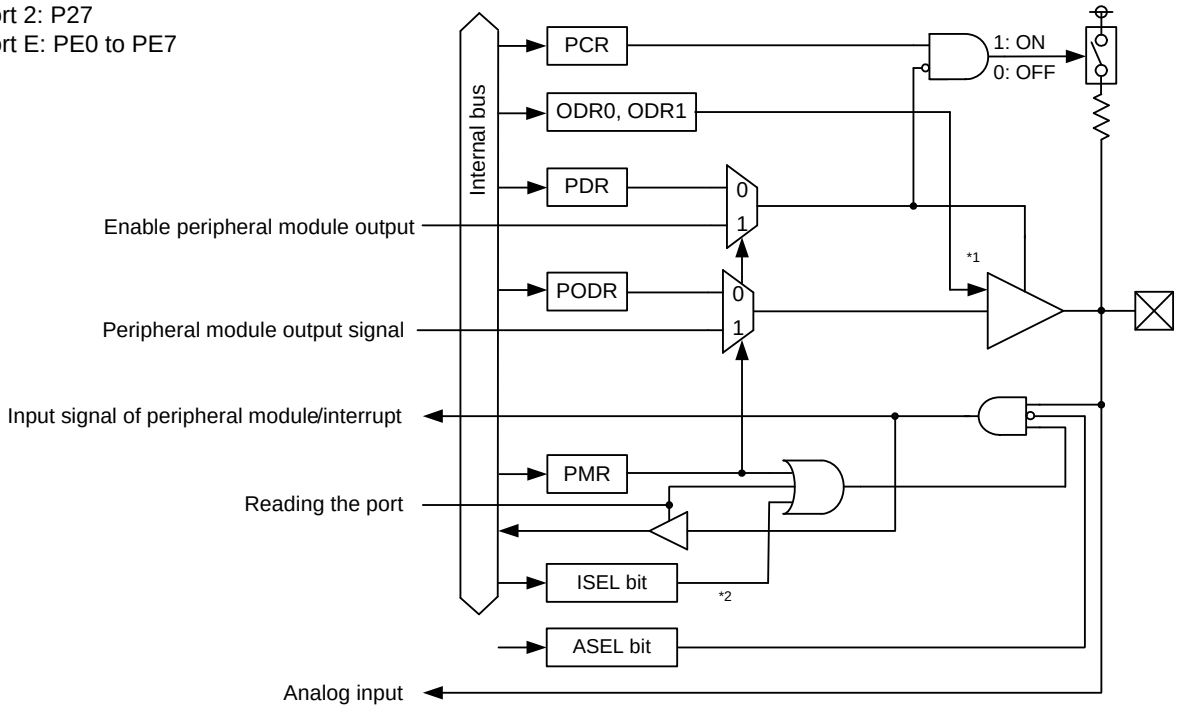

**Figure 18.1 I/O Port Configuration (1)**

Port 1: P14 to P17
Port 2: P26
Port 3: P30 to P32
Port A: PA0, PA1, PA3, PA4, PA6
Port B: PB0, PB1, PB3, PB5 to PB7
Port C: PC0 to PC7




Note 1. Control signal for N-channel open-drain output (other than P14)
Control signal for N-channel open-drain and P-channel open-drain (P14)
Note 2. An external interrupt function is multiplexed on this pin

Port 2: P27
Port E: PE0 to PE7




Note 1. Control signal for N-channel open-drain output (other than PE1)
Control signal for N-channel open-drain and P-channel open-drain (PE1)
Note 2. An external interrupt function is multiplexed on this pin

**Figure 18.2 I/O Port Configuration (2)**

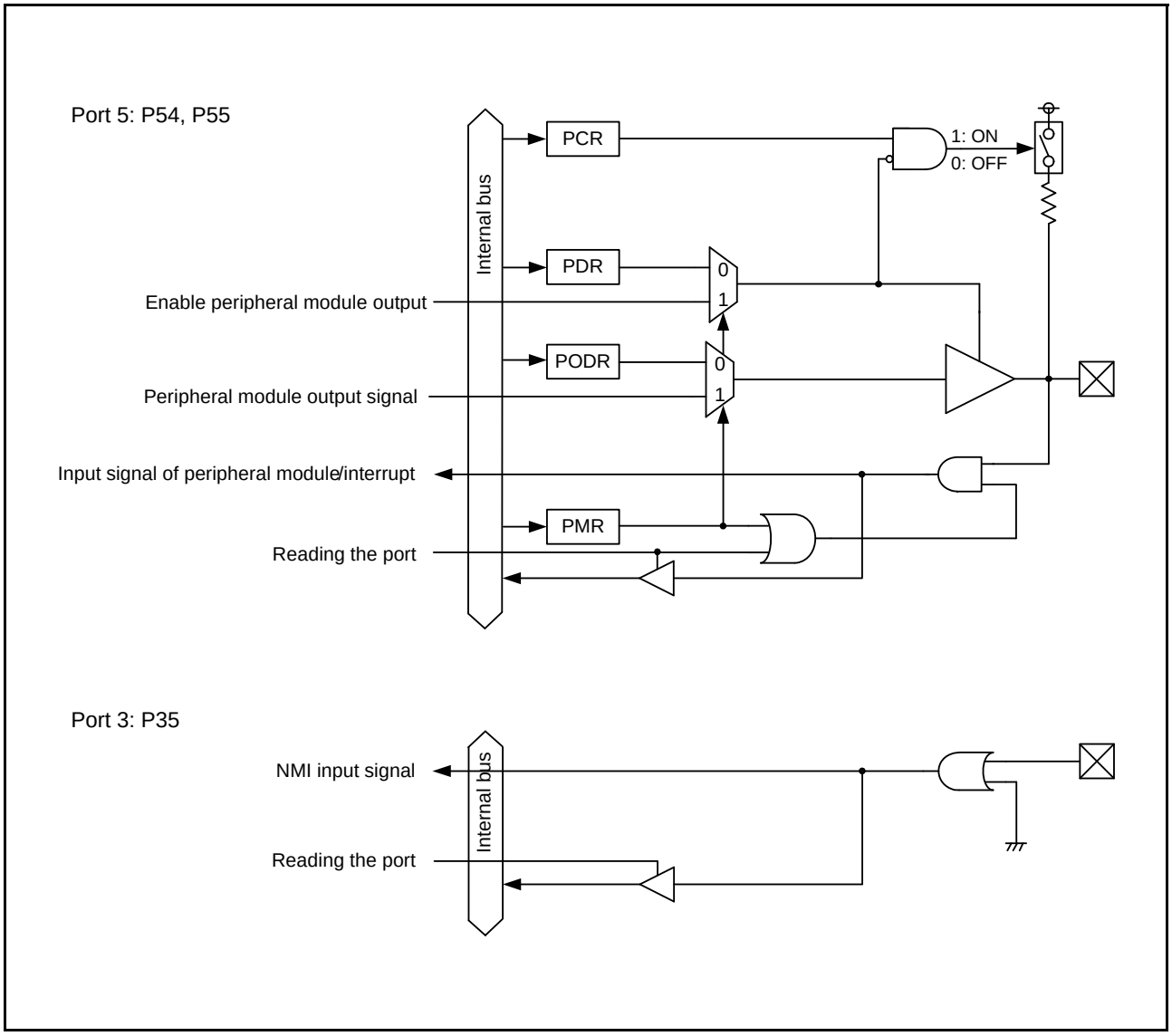



**Figure 18.3 I/O Port Configuration (3)**

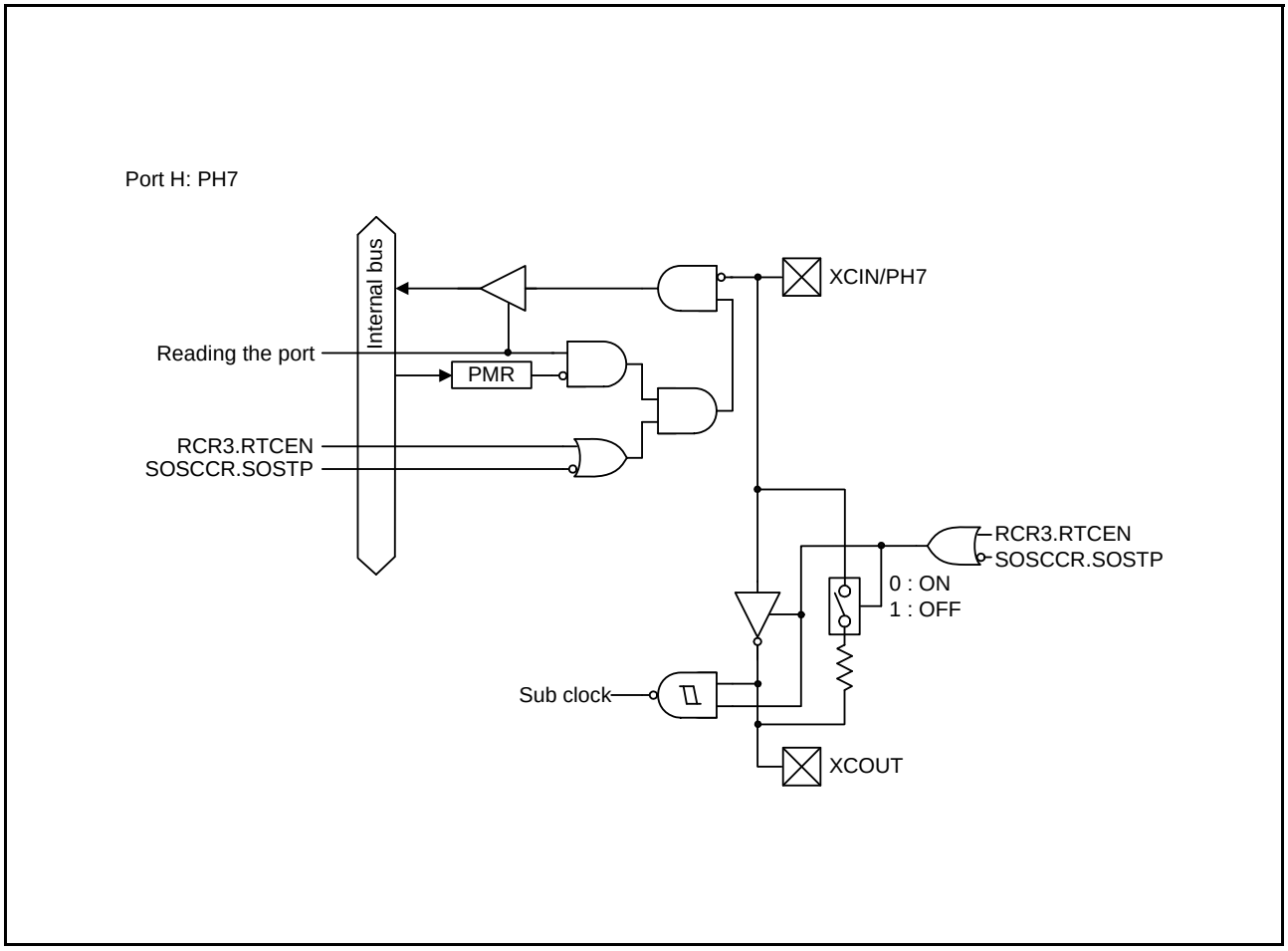



**Figure 18.4 I/O Port Configuration (4)**

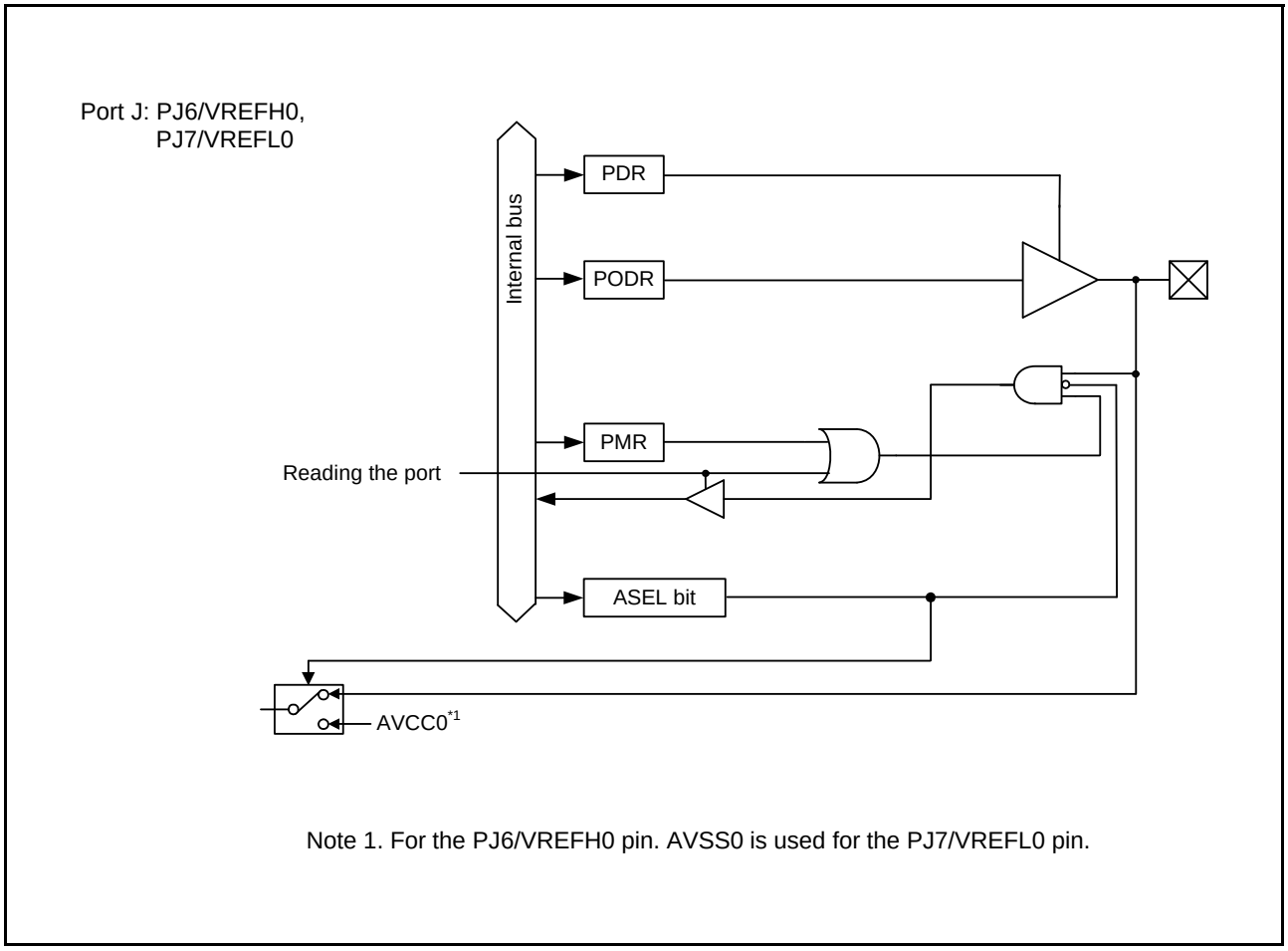



**Figure 18.5 I/O Port Configuration (5)**

## 18.3 Register Descriptions

### 18.3.1 Port Direction Register (PDR)

Address(es): PORT0.PDR: 0008 C000h, PORT1.PDR: 0008 C001h, PORT2.PDR: 0008 C002h, PORT3.PDR: 0008 C003h, PORT4.PDR: 0008 C004h, PORT5.PDR: 0008 C005h, PORTA.PDR: 0008 C00Ah, PORTB.PDR: 0008 C00Bh, PORTC.PDR: 0008 C00Ch, PORTE.PDR: 0008 C00Eh, PORTJ.PDR: 0008 C012h

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | B7 | B6 | B5 | B4 | B3 | B2 | B1 | B0 |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b0 | B0 | Pm0 Direction Control | 0: Input (Functions as an input pin.)<br>1: Output (Functions as an output pin.) | R/W |
| b1 | B1 | Pm1 Direction Control | | R/W |
| b2 | B2 | Pm2 Direction Control | | R/W |
| b3 | B3 | Pm3 Direction Control | | R/W |
| b4 | B4 | Pm4 Direction Control | | R/W |
| b5 | B5 | Pm5 Direction Control | | R/W |
| b6 | B6 | Pm6 Direction Control | | R/W |
| b7 | B7 | Pm7 Direction Control | | R/W |

m = 0 to 5, A to C, E, J

PDR is used to select the input or output direction for individual pins of the corresponding port m when the pins are configured as the general I/O pins.

Each bit of PORTm.PDR corresponds to each pin of port m; I/O direction can be specified in 1-bit units. For software compatibility, 1 (output) can be written to the bits corresponding to port m on the 64-pin product but which do not exist on a product with fewer than 64 pins are reserved.

The PORT3.PDR.B5 bit is reserved, because the P35 pin is input only. The bit corresponding to a pin that does not exist is also reserved. A reserved bit is read as 0. The write value should be 0.

## 18.3.2 Port Output Data Register (PODR)

Address(es): PORT0.PODR: 0008 C020h, PORT1.PODR: 0008 C021h, PORT2.PODR: 0008 C022h, PORT3.PODR: 0008 C023h, PORT4.PODR: 0008 C024h, PORT5.PODR: 0008 C025h, PORTA.PODR: 0008 C02Ah, PORTB.PODR: 0008 C02Bh, PORTC.PODR: 0008 C02Ch, PORTE.PODR: 0008 C02Eh, PORTJ.PODR: 0008 C032h

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | B7 | B6 | B5 | B4 | B3 | B2 | B1 | B0 |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b0 | B0 | Pm0 Output Data Store | Holds output data. | R/W |
| b1 | B1 | Pm1 Output Data Store | | R/W |
| b2 | B2 | Pm2 Output Data Store | | R/W |
| b3 | B3 | Pm3 Output Data Store | | R/W |
| b4 | B4 | Pm4 Output Data Store | | R/W |
| b5 | B5 | Pm5 Output Data Store | | R/W |
| b6 | B6 | Pm6 Output Data Store | | R/W |
| b7 | B7 | Pm7 Output Data Store | | R/W |

m = 0 to 5, A to C, E, J

PODR holds the data to be output from the pins used for general output ports.
Bits corresponding to port m on the 64 pin-product but which do not exist on a product with fewer than 64 pins are reserved. Write 0 to these bits.
The PORT3.PODR.B5 bit is reserved, because the P35 pin is input only. The bit corresponding to a pin that does not exist is reserved. A reserved bit is read as 0. The write value should be 0.

### 18.3.3 Port Input Data Register (PIDR)

Address(es): PORT0.PIDR: 0008 C040h, PORT1.PIDR: 0008 C041h, PORT2.PIDR: 0008 C042h, PORT3.PIDR: 0008 C043h, PORT4.PIDR: 0008 C044h, PORT5.PIDR: 0008 C045h, PORTA.PIDR: 0008 C04Ah, PORTB.PIDR: 0008 C04Bh, PORTC.PIDR: 0008 C04Ch, PORTE.PIDR: 0008 C04Eh, PORTH.PIDR: 0008 C051h, PORTJ.PIDR: 0008 C052h

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | B7 | B6 | B5 | B4 | B3 | B2 | B1 | B0 |
| Value after reset: | x | x | x | x | x | x | x | x |

x: Undefined

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b0 | B0 | Pm0 | Indicates individual pin states of the corresponding port. | R/W |
| b1 | B1 | Pm1 | | R/W |
| b2 | B2 | Pm2 | | R/W |
| b3 | B3 | Pm3 | | R/W |
| b4 | B4 | Pm4 | | R/W |
| b5 | B5 | Pm5 | | R/W |
| b6 | B6 | Pm6 | | R/W |
| b7 | B7 | Pm7 | | R/W |

m = 0 to 5, A to C, E, H, J

PIDR indicates individual pin states of port m.
The pin states of port m can be read with the PORTm.PIDR, regardless of the values of PORTm.PDR and PORTm.PMR.
The NMI pin state is reflected in the P35 bit.
The bit corresponding to a pin that does not exist is reserved. A reserved bit is read as undefined and cannot be modified.

**Notes when using PH7 as a general input port and PJ6 and PJ7 as general I/O ports**

When using PH7 as a general input port, follow the procedure below to make settings.

1. Set the sub-clock oscillator control bit in RTC control register 3 (RCR3.RTCEN) to 0. For details on this register, refer to section 23.2.19, RTC Control Register 3 (RCR3).
2. Set the sub-clock oscillator stop bit in the sub-clock oscillator control register (SOSCCR.SOSTP) to 1. For details on the functions and rewriting of this register, refer to section 9.2.6, Sub-Clock Oscillator Control Register (SOSCCR).

When using PJ6 and PJ7 as a general input port, follow the procedure below to make settings.

1. Set the PJ6PFS.ASEL bit to 0. (When using the PJ6 port)
   Set the PJ7PFS.ASEL bit to 0. (When using the PJ7 port)
2. Set the PORTJ.PMR.B6 to 0. (When using the PJ6 port)
   Set the PORTJ.PMR.B7 to 0. (When using the PJ7 port)

## 18.3.4 Port Mode Register (PMR)

Address(es): PORT0.PMR: 0008 C060h, PORT1.PMR: 0008 C061h, PORT2.PMR: 0008 C062h, PORT3.PMR: 0008 C063h, PORT4.PMR: 0008 C064h, PORT5.PMR: 0008 C065h, PORTA.PMR: 0008 C06Ah, PORTB.PMR: 0008 C06Bh, PORTC.PMR: 0008 C06Ch, PORTE.PMR: 0008 C06Eh, PORTH.PMR: 0008 C071h, PORTJ.PMR: 0008 C072h

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | B7 | B6 | B5 | B4 | B3 | B2 | B1 | B0 |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b0 | B0 | Pm0 Pin Mode Control | 0: Use pin as general I/O port.<br>1: Use pin as I/O port for peripheral functions. | R/W |
| b1 | B1 | Pm1 Pin Mode Control | | R/W |
| b2 | B2 | Pm2 Pin Mode Control | | R/W |
| b3 | B3 | Pm3 Pin Mode Control | | R/W |
| b4 | B4 | Pm4 Pin Mode Control | | R/W |
| b5 | B5 | Pm5 Pin Mode Control | | R/W |
| b6 | B6 | Pm6 Pin Mode Control | | R/W |
| b7 | B7 | Pm7 Pin Mode Control | | R/W |

m = 0 to 5, A to C, E, H, J

Each bit of PORTm.PMR corresponds to each pin of port m; pin function can be specified in 1-bit units. Bits corresponding to port m on the 64 pin-product but which do not exist on a product with fewer than 64 pins are reserved. Write 0 to these bits.

The bit corresponding to a pin that does not exist is reserved. A reserved bit is read as 0. The write value should be 0.

## 18.3.5 Open Drain Control Register 0 (ODR0)

Address(es): PORT3.ODR0: 0008 C086h, PORTA.ODR0: 0008 C094h, PORTB.ODR0: 0008 C096h, PORTC.ODR0: 0008 C098h, PORTE.ODR0: 0008 C09Ch

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | B7 | B6 | B5 | B4 | B3 | B2 | B1 | B0 |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b0 | B0 | Pm0 Output Type Select | 0: CMOS output<br>1: N-channel open-drain output | R/W |
| b1 | B1 | Reserved | This bit is read as 0. The write value should be 0. | R/W |
| b2 | B2 | Pm1 Output Type Select | • P31, PA1, PB1<br>b2 0: CMOS output<br>1: N-channel open-drain output<br>b3 These bits are read as 0. The write value should be 0.<br>• PE1<br>b3 b2<br>0 0: CMOS output<br>0 1: N-channel open-drain output<br>1 0: P-channel open-drain output<br>1 1: Setting prohibited. | R/W |
| b3 | B3 | | | R/W |
| b4 | B4 | Pm2 Output Type Select | 0: CMOS output<br>1: N-channel open-drain output | R/W |
| b5 | B5 | Reserved | This bit is read as 0. The write value should be 0. | R/W |
| b6 | B6 | Pm3 Output Type Select | 0: CMOS output<br>1: N-channel open-drain output | R/W |
| b7 | B7 | Reserved | This bit is read as 0. The write value should be 0. | R/W |

m = 3, A to C, E

Bits corresponding to port m on the 64 pin-product but which do not exist on a product with fewer than 64 pins are reserved. Write 0 to these bits.

The bit corresponding to a pin that does not exist is reserved. A reserved bit is read as 0. The write value should be 0.

## 18.3.6 Open Drain Control Register 1 (ODR1)

Address(es): PORT1.ODR1: 0008 C083h, PORT2.ODR1: 0008 C085h, PORTA.ODR1: 0008 C095h, PORTB.ODR1: 0008 C097h, PORTC.ODR1: 0008 C099h, PORTE.ODR1: 0008 C09Dh

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | B7 | B6 | B5 | B4 | B3 | B2 | B1 | B0 |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b0 | B0 | Pm4 Output Type Select | • PA4, PC4, PE4<br>b2 0: CMOS output<br>1: N-channel open-drain output<br>b3 These bits are read as 0. The write value should be 0.<br>• P14<br>b3 b2<br>0 0: CMOS output<br>0 1: N-channel open-drain output<br>1 0: P-channel open-drain output<br>1 1: Setting prohibited. | R/W |
| b1 | B1 | | | R/W |
| b2 | B2 | Pm5 Output Type Select | 0: CMOS output<br>1: N-channel open-drain output | R/W |
| b3 | B3 | Reserved | This bit is read as 0. The write value should be 0. | R/W |
| b4 | B4 | Pm6 Output Type Select | 0: CMOS output<br>1: N-channel open-drain output | R/W |
| b5 | B5 | Reserved | This bit is read as 0. The write value should be 0. | R/W |
| b6 | B6 | Pm7 Output Type Select | 0: CMOS output<br>1: N-channel open-drain output | R/W |
| b7 | B7 | Reserved | This bit is read as 0. The write value should be 0. | R/W |

m = 1 to 2, A to C, E

Bits corresponding to port m on the 64 pin-product but which do not exist on a product with fewer than 64 pins are reserved. Write 0 to these bits.

The PORT3.ODR1.B2 bit is reserved, because the P35 pin is input only. The bit corresponding to a pin that does not exist is reserved. A reserved bit is read as 0. The write value should be 0.

## 18.3.7 Pull-Up Control Register (PCR)

Address(es): PORT0.PCR: 0008 C0C0h, PORT1.PCR: 0008 C0C1h, PORT2.PCR: 0008 C0C2h, PORT3.PCR: 0008 C0C3h, PORT5.PCR: 0008 C0C5h, PORTA.PCR: 0008 C0CAh, PORTB.PCR: 0008 C0CBh, PORTC.PCR: 0008 C0CCh, PORTE.PCR: 0008 C0CEh

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | B7 | B6 | B5 | B4 | B3 | B2 | B1 | B0 |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b0 | B0 | Pm0 Input Pull-Up Resistor Control | 0: Disables an input pull-up resistor.<br>1: Enables an input pull-up resistor. | R/W |
| b1 | B1 | Pm1 Input Pull-Up Resistor Control | | R/W |
| b2 | B2 | Pm2 Input Pull-Up Resistor Control | | R/W |
| b3 | B3 | Pm3 Input Pull-Up Resistor Control | | R/W |
| b4 | B4 | Pm4 Input Pull-Up Resistor Control | | R/W |
| b5 | B5 | Pm5 Input Pull-Up Resistor Control | | R/W |
| b6 | B6 | Pm6 Input Pull-Up Resistor Control | | R/W |
| b7 | B7 | Pm7 Input Pull-Up Resistor Control | | R/W |

m = 0 to 3, 5, A to C, E

While a pin is in the input state with the corresponding bit in PORTm.PCR set to 1, the pull-up resistor connected to the pin is enabled.

When a pin is set as a general port output pin, or a peripheral function output pin, the pull-up resistor for the pin is disabled regardless of the settings of PCR.

The pull-up resistor is also disabled in the reset state.

The PORT3.PCR.B5 bit is reserved. The bit corresponding to a pin that does not exist is reserved. A reserved bit is read as 0. The write value should be 0.

### 18.3.8 Port Switching Register A (PSRA)

Address(es): PORT.PSRA 0008 C121h

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | PSEL7 | PSEL6 | — | — | — | — | — | — |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b5 to b0 | — | Reserved | This bit is read as 0. The write value should be 0. | R/W |
| b6 | PSEL6 | PB6/PC0 Switching | 0: PB6 general I/O port function is selected<br>1: PC0 general I/O port function is selected | R/W |
| b7 | PSEL7 | PB7/PC1 Switching | 0: PB7 general I/O port function is selected<br>1: PC1 general I/O port function is selected | R/W |

The PSRA register is for 64-pin packages. The PSRA register is used to select either the general I/O functions of PB6 and PB7 or those of PC0 and PC1. When 1 is written to the PSEL6 and PSEL7 bits, port C can be used as an 8-bit port. As for the I/O functions of the peripheral functions, functions multiplexed with PB6 and PB7 are enabled. To enable the peripheral functions, write 1 to the corresponding pin mode control bit of the PORTB.PMR register.
Rewriting to this register must be performed when the PMR, PDR, and PCR registers for the corresponding pins are 0.

## 18.3.9 Port Switching Register B (PSRB)

Address(es): PORT.PSRB 0008 C120h

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | PSEL5 | — | PSEL3 | — | PSEL1 | PSEL0 |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b0 | PSEL0 | PB0/PC0 Switching | 0: PB0 general I/O port function is selected<br>1: PC0 general I/O port function is selected | R/W |
| b1 | PSEL1 | PB1/PC1Switching | 0: PB1 general I/O port function is selected<br>1: PC1 general I/O port function is selected | R/W |
| b2 | — | Reserved | This bit is read as 0. The write value should be 0. | R/W |
| b3 | PSEL3 | PB3/PC2 Switching | 0: PB3 general I/O port function is selected<br>1: PC2 general I/O port function is selected | R/W |
| b4 | — | Reserved | This bit is read as 0. The write value should be 0. | R/W |
| b5 | PSEL5 | PB5/PC3 Switching | 0: PB5 general I/O port function is selected<br>1: PC3 general I/O port function is selected | R/W |
| b7, b6 | — | Reserved | This bit is read as 0. The write value should be 0. | R/W |

The PSRB register is for 48-pin packages. The PSRB register is used to select either the general I/O functions of PB0, PB1, PB3, and PB5 and those of PC0 to PC3. When 1 is written to the PSEL0, PSEL1, PSEL3, and PSEL5 bits, port C can be used as an 8-bit port.

As for the I/O functions of the peripheral functions, functions multiplexed with PB0, PB1, PB3, and PB5 are enabled. To enable the peripheral functions, write 1 to the corresponding pin mode control bit of the PORTB.PMR register.

Rewriting to this register must be performed when the PMR, PDR, and PCR registers for the corresponding pins are 0.

## 18.4 Handling of Unused Pins

The handling of unused pins is listed in Table 18.3.

**Table 18.3 Handling of Unused Pins**

| Pin Name | Description |
|---|---|
| MD | (Always used as mode pins) |
| RES# | Connect this pin to VCC via a pull-up resistor. |
| P35/NMI | Connect this pin to VCC via a pull-up resistor. |
| USB0_DM, USB0_DP | Leave this pin open. |
| EXTAL | Connect this pin to VSS via a pull-down resistor. |
| XTAL | Leave this pin open. |
| PH7/XCIN | When the sub-clock is not used, set the RCR3.RTCEN bit to 0 and the SOSCCR.SOSTP bit to 1 (general port PH7).<br>When this pin is not also used as port PH7, handle it in the same way as the input setting of ports 0 to 5, A to C, E, H, and J. |
| XCOUT | Leave this pin open. |
| Port 0 to 5, A to C, E, H, J (for pins that exist on products with fewer than 64 pins) | • Set these pins to input (PORTn.PDR bit = 0) and connect each of them to VCC via a pull-up resistor or to VSS via a pull-down resistor. *1<br>• Set these pins to output (PORTn.PDR bit = 1), set the output data to 0 (PORTn.PODR bit = 0), and leave them open. *1, *2 |
| Port 0 to 5, A to C, E, H, J (for pins that do not exist on products with fewer than 64 pins) | Set these pins to output (PORTn.PDR bit = 1), set the output data to 0 (PORTn.PODR bit = 0), and leave them open. *1, *2 |
| VREFH0 | When this pin is not used as VREFH0, set the PJ6PFS.ASEL bit to 0 (general port PJ6).<br>When this pin is not also used as port PJ6, handle it in the same way as the handling of ports 0 to 5, A to C, E, H, and J. |
| VREFL0 | When this pin is not used as VREFL0, set the PJ7PFS.ASEL bit to 0 (general port PJ7).<br>When this pin is not also used as port PJ7, handle it in the same way as the handling of ports 0 to 5, A to C, E, H, and J. |

Note 1. Set the PORTn.PMR bit to 0 and the PmnPFS.ISEL and ASEL bits to 0.
Note 2. If these ports are set to output mode and left open, they remain in input mode after the reset is released and before they are switched to output mode. The voltage level of these pins may be unstable and the power current may increase while the ports remain in input mode.

# 19. Multi-Function Pin Controller (MPC)

## 19.1 Overview

The multi-function pin controller (MPC) is used to allocate input and output signals for peripheral modules and input interrupt signals to pins from among multiple ports.

Table 19.1 shows the allocation of pin functions to multiple pins. A and N/A in the table indicate whether the pins are available or not available on the given package. Allocating the same function to more than one pin is prohibited.

**Table 19.1 Allocation of Pin Functions to Multiple Pins (1/5)**

| | | | | Package | | | |
|---|---|---|---|---|---|---|---|
| **Module/Function** | **Channel** | **Pin Functions** | **Allocation Port** | **64-pin** | **48-pin** | **40-pin** | **36-pin** |
| Interrupt | | NMI (input) | P35 | A | A | A | A |
| Interrupt | IRQ0 | IRQ0 (input) | P30 | A | N/A | N/A | N/A |
| | | | PE0 | A | A | A | A |
| | IRQ1 | IRQ1 (input) | P31 | A | N/A | N/A | N/A |
| | | | PE1 | A | A | A | A |
| | IRQ2 | IRQ2 (input) | P32 | A | N/A | A | N/A |
| | | | PB0 | A | A | A | A |
| | | | PC4 | A | A | A | A |
| | IRQ3 | IRQ3 (input) | P27 | A | A | A | A |
| | | | PE3 | A | A | A | A |
| | | | PA6 | A | A | A | A |
| | IRQ4 | IRQ4 (input) | P14 | A | A | A | A |
| | | | PB1 | A | A | N/A | N/A |
| | | | PE4 | A | A | A | A |
| | IRQ5 | IRQ5 (input) | P15 | A | A | A | A |
| | | | PA4 | A | A | A | A |
| | | | PE5 | A | N/A | N/A | N/A |
| | IRQ6 | IRQ6 (input) | P16 | A | A | A | A |
| | | | PA3 | A | A | A | A |
| | | | PE6 | A | N/A | N/A | N/A |
| | IRQ7 | IRQ7 (input) | P17 | A | A | A | A |
| | | | PE2 | A | A | A | A |
| | | | PE7 | A | A | N/A | N/A |
| Multi-function timer unit 2 | MTU0 | MTIOC0A (I/O) | P14 | A | A | A | A |
| | | | PB3 | A | A | A | A |
| | | | PE3 | A | A | A | A |
| | | MTIOC0B (I/O) | P15 | A | A | A | A |
| | | | PA1 | A | A | A | N/A |
| | | MTIOC0C (I/O) | P17 | A | A | A | A |
| | | | P32 | A | N/A | A | N/A |
| | | | PB0 | A | A | A | A |
| | | | PB1 | A | A | N/A | N/A |
| | | MTIOC0D (I/O) | PA3 | A | A | A | A |

**Table 19.1 Allocation of Pin Functions to Multiple Pins (2/5)**

| Module/Function | Channel | Pin Functions | Allocation Port | Package | | | |
|---|---|---|---|---|---|---|---|
| | | | | 64-pin | 48-pin | 40-pin | 36-pin |
| Multi-function timer unit 2 | MTU1 | MTIOC1A (I/O) | PE4 | A | A | A | A |
| | | MTIOC1B (I/O) | PA3 | A | A | A | A |
| | | | PB5 | A | A | N/A | N/A |
| | | | PE3 | A | A | A | A |
| | MTU2 | MTIOC2A (I/O) | P26 | A | A | A | N/A |
| | | | PA6 | A | A | A | A |
| | | | PB5 | A | A | N/A | N/A |
| | | | PE0 | A | A | A | A |
| | | MTIOC2B (I/O) | P27 | A | A | A | A |
| | | | PA4 | A | A | A | A |
| | | | PE5 | A | N/A | N/A | N/A |
| | MTU3 | MTIOC3A (I/O) | P14 | A | A | A | A |
| | | | P17 | A | A | A | A |
| | | | PC7 | A | A | N/A | N/A |
| | | | PE4 | A | A | A | A |
| | | MTIOC3B (I/O) | P17 | A | A | A | A |
| | | | PB3 | A | A | A | A |
| | | | PB7 | A | N/A | N/A | N/A |
| | | | PC5 | A | A | N/A | N/A |
| | | MTIOC3C (I/O) | P16 | A | A | A | A |
| | | | PC6 | A | A | N/A | N/A |
| | | MTIOC3D (I/O) | P16 | A | A | A | A |
| | | | PB6 | A | N/A | N/A | N/A |
| | | | PC4 | A | A | A | A |
| | MTU4 | MTIOC4A (I/O) | PA0 | A | N/A | N/A | N/A |
| | | | PB3 | A | A | A | A |
| | | | PE2 | A | A | A | A |
| | | MTIOC4B (I/O) | P30 | A | N/A | N/A | N/A |
| | | | P54 | A | N/A | N/A | N/A |
| | | | PC2 | A | N/A | N/A | N/A |
| | | | PE3 | A | A | A | A |
| | | MTIOC4C (I/O) | PB1 | A | A | N/A | N/A |
| | | | PE1 | A | A | A | A |
| | | | PE5 | A | N/A | N/A | N/A |
| | | MTIOC4D (I/O) | P31 | A | N/A | N/A | N/A |
| | | | P55 | A | N/A | N/A | N/A |
| | | | PC3 | A | N/A | N/A | N/A |
| | | | PE4 | A | A | A | A |
| | MTU5 | MTIC5U (input) | PA4 | A | A | A | A |
| | | MTIC5V (input) | PA6 | A | A | A | A |
| | | MTIC5W (input) | PB0 | A | A | A | A |

**Table 19.1 Allocation of Pin Functions to Multiple Pins (3/5)**

| Module/Function | Channel | Pin Functions | Allocation Port | Package | | | |
|---|---|---|---|---|---|---|---|
| | | | | 64-pin | 48-pin | 40-pin | 36-pin |
| Multi-function timer unit 2 | MTU | MTCLKA (input) | P14 | A | A | A | A |
| | | | PA4 | A | A | A | A |
| | | | PC6 | A | A | N/A | N/A |
| | | MTCLKB (input) | P15 | A | A | A | A |
| | | | PA6 | A | A | A | A |
| | | | PC7 | A | A | N/A | N/A |
| | | MTCLKC (input) | PA1 | A | A | A | N/A |
| | | | PC4 | A | A | A | A |
| | | MTCLKD (input) | PA3 | A | A | A | A |
| | | | PC5 | A | A | N/A | N/A |
| Port output enable 2 | POE0 | POE0# (input) | PC4 | A | A | A | A |
| | | | PA3 | A | A | A | A |
| | POE1 | POE1# (input) | PB5 | A | A | N/A | N/A |
| | POE2 | POE2# (input) | PA6 | A | A | A | A |
| | POE3 | POE3# (input) | PB3 | A | A | A | A |
| | | | PE0 | A | A | A | A |
| | POE8 | POE8# (input) | P17 | A | A | A | A |
| | | | P30 | A | N/A | N/A | N/A |
| | | | PE3 | A | A | A | A |
| Serial communications interface | SCI1 | RXD1 (input)/ SMISO1 (I/O)/ SSCL1 (I/O) | P15 | A | A | A | A |
| | | | P30 | A | N/A | N/A | N/A |
| | | | PC6 | A | A | N/A | N/A |
| | | TXD1 (output)/ SMOSI1 (I/O)/ SSDA1 (I/O) | P16 | A | A | A | A |
| | | | P26 | A | A | A | N/A |
| | | | PC7 | A | A | N/A | N/A |
| | | SCK1 (I/O) | P17 | A | A | A | A |
| | | | P27 | A | A | A | A |
| | | | PC5 | A | A | N/A | N/A |
| | | CTS1# (input)/ RTS1# (output) SS1# (input) | P14 | A | A | A | A |
| | | | P31 | A | N/A | N/A | N/A |
| | SCI5 | RXD5 (input)/ SMISO5 (I/O)/ SSCL5 (I/O) | PA3 | A | A | A | A |
| | | | PC2 | A | N/A | N/A | N/A |
| | | TXD5 (output)/ SMOSI5 (I/O)/ SSDA5 (I/O) | PA4 | A | A | A | A |
| | | | PC3 | A | N/A | N/A | N/A |
| | | SCK5 (I/O) | PA1 | A | A | A | N/A |
| | | | PC4 | A | A | A | A |
| | | CTS5# (input)/ RTS5# (output)/ SS5# (input) | PA6 | A | A | A | A |

**Table 19.1 Allocation of Pin Functions to Multiple Pins (4/5)**

| Module/Function | Channel | Pin Functions | Allocation Port | Package | | | |
|---|---|---|---|---|---|---|---|
| | | | | 64-pin | 48-pin | 40-pin | 36-pin |
| Serial communications interface | SCI12SDA0 | SCK12 (I/O) | PE0 | A | A | A | A |
| | | | P27 | A | A | A | A |
| | | RXD12 (input)/ SMISO12 (I/O)/ SSCL12 (I/O)/ RXDX12 (input) | PE2 | A | A | A | A |
| | | | P17 | A | A | A | A |
| | | TXD12 (output)/ SMOSI12 (I/O)/ SSDA12 (I/O)/ TXDX12 (output)/ SIOX12 (I/O) | PE1 | A | A | A | A |
| | | | P14 | A | A | A | A |
| | | CTS12# (input)/ RTS12# (output)/ SS12# (input) | PE3 | A | A | A | A |
| I²C bus interface | RIIC0 | SCL0 (I/O) | P16 | A | A | A | A |
| | | | PB0 | A | A | A | A |
| | | SDA0 (I/O) | P17 | A | A | A | A |
| | | | PA6 | A | A | A | A |
| Serial peripheral interface | RSPI0 | RSPCKA (I/O) | P15 | A | A | A | A |
| | | | PB0 | A | A | A | A |
| | | | PC5 | A | A | N/A | N/A |
| | | | PE3 | A | A | A | A |
| | | MOSIA (I/O) | P16 | A | A | A | A |
| | | | PA6 | A | A | A | A |
| | | | PE4 | A | A | A | A |
| | | | PC6 | A | A | N/A | N/A |
| | | MISOA (I/O) | P17 | A | A | A | A |
| | | | PC7 | A | A | N/A | N/A |
| | | | PA3 | A | A | A | A |
| | | SSLA0 (I/O) | P14 | A | A | A | A |
| | | | PA4 | A | A | A | A |
| | | | PC4 | A | A | A | A |
| | | SSLA1 (output) | PA0 | A | N/A | N/A | N/A |
| | | SSLA2 (output) | PA1 | A | A | A | N/A |
| | | SSLA3 (output) | PC2 | A | N/A | N/A | N/A |
| USB 2.0 host/function module | USB0 | USB0_EXICEN (output) | PC6 | A | A | N/A | N/A |
| | | USB0_VBUSEN (output) | P16 | A | A | A | A |
| | | | PC4 | A | A | A | A |
| | | | P26 | A | A | A | N/A |
| | | USB0_OVRCURA (input) | P14 | A | A | A | A |
| | | | PB3 | A | A | A | A |
| | | USB0_OVRCURB (input) | P16 | A | A | A | A |
| | | | PC7 | A | A | N/A | N/A |
| | | USB0_ID (input) | PC5 | A | A | N/A | N/A |
| | | USB0_VBUS (input)*2 | P16 | A | A | A | A |
| | | USB0_VBUS (input)*3 | PC4 | A | A | A | A |

**Table 19.1 Allocation of Pin Functions to Multiple Pins (5/5)**

| Module/Function | Channel | Pin Functions | Allocation Port | Package | | | |
|---|---|---|---|---|---|---|---|
| | | | | 64-pin | 48-pin | 40-pin | 36-pin |
| Realtime clock | | RTCOUT (output) | P16 | A | A | N/A | N/A |
| | | | P32 | A | N/A | N/A | N/A |
| | | | PB0 | A | A | N/A | N/A |
| | | | PA1 | A | A | N/A | N/A |
| 12-bit A/D converter | | AN000 (input)*1 | P40 | A | A | N/A | N/A |
| | | AN001 (input)*1 | P41 | A | A | A | A |
| | | AN002 (input)*1 | P42 | A | A | A | A |
| | | AN003 (input)*1 | P43 | A | N/A | N/A | N/A |
| | | AN004 (input)*1 | P44 | A | N/A | N/A | N/A |
| | | AN006 (input)*1 | P46 | A | A | A | N/A |
| | | AN008 (input)*1 | PE0 | A | A | A | A |
| | | AN009 (input)*1 | PE1 | A | A | A | A |
| | | AN010 (input)*1 | PE2 | A | A | A | A |
| | | AN011 (input)*1 | PE3 | A | A | A | A |
| | | AN012 (input)*1 | PE4 | A | A | A | A |
| | | AN013 (input)*1 | PE5 | A | N/A | N/A | N/A |
| | | AN014 (input)*1 | PE6 | A | N/A | N/A | N/A |
| | | AN015 (input)*1 | PE7 | A | A | N/A | N/A |
| | | VREFH0 (input) | PJ6 | A | A | A | A |
| | | VREFL0 (input) | PJ7 | A | A | A | A |
| | | ADTRG0# (input) | P16 | A | A | A | A |
| | | | P27 | A | A | A | A |
| | | | PB0 | A | A | A | A |
| D/A converter | | DA0 (output)*1 | P03 | A | N/A | N/A | N/A |
| | | DA1 (output)*1 | P05 | A | N/A | N/A | N/A |
| Clock | | CLKOUT (output) | P15 | A | A | A | A |
| | | | PC4 | A | A | A | A |
| Clock frequency accuracy measurement circuit | | CACREF (input) | P27 | A | A | A | A |
| | | | PA0 | A | N/A | N/A | N/A |
| | | | PC7 | A | A | N/A | N/A |
| Voltage detection circuit | | CMPA2 (input)*1 | P27 | A | A | A | A |

Note 1. Select general input (by setting the Bm bits for the given pin in the PDR and PMR for the given port to 0) for the pin if this pin function is to be used.
Note 2. 5 V tolerant.
Note 3. Not 5 V tolerant.

## 19.2 Register Descriptions

Registers and bits for pins that are not present due to differences according to the package are reserved. Write the value after a reset when writing to such bits.

### 19.2.1 Write-Protect Register (PWPR)

Address(es): 0008 C11Fh

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | B0WI | PFSWE | — | — | — | — | — | — |
| Value after reset: | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b5 to b0 | — | Reserved | These bits are read as 0. The write value should be 0. | — |
| b6 | PFSWE | PFS Register Write Enable | 0: Writing to the PFS register is disabled<br>1: Writing to the PFS register is enabled | R/W |
| b7 | B0WI | PFSWE Bit Write Disable | 0: Writing to the PFSWE bit is enabled<br>1: Writing to the PFSWE bit is disabled | R/W |

**PFSWE Bit (PFS Register Write Enable)**

Writing to the PmnPFS register is enabled only when the PFSWE bit is set to 1.
To set the PFSWE bit to 1, write 1 to the PFSWE bit after writing 0 to the B0WI bit.

**B0WI Bit (PFSWE Bit Write Disable)**

Writing to the PFSWE bit is enabled only when the B0WI bit is set to 0.

## 19.2.2 P0n Pin Function Control Register (P0nPFS) (n = 3, 5)

Address(es): P03PFS: 0008 C143h, P05PFS: 0008 C145h

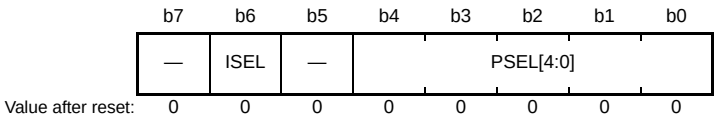
| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | ASEL | — | — | — | — | — | — | — |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b6 to b0 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |
| b7 | ASEL | Analog Function Select | 0: Used as other than an analog pin<br>1: Used as an analog pin<br>P03: DA0 (64 pins)<br>P05: DA1 (64 pins) | R/W |

The Pmn pin function control register (PmnPFS) selects the pin function.
Bits PSEL[4:0] select the peripheral function assigned to each pin.
The ISEL bit is set when a pin is used as an IRQ input pin. This setting can be used with the combination of the peripheral function, though IRQn (external pin interrupt) of the same number should not be enabled by two or more pins.
The ASEL bit is set when a pin is used as an analog pin. When switching a pin to analog using the ASEL bit, set the corresponding port mode register bit (PORTm.PMR) to “general I/O port” and the port direction register bit (PORTm.PDR) to “input”. The pin state cannot be read at this point. The PmnPFS register is protected by the write protect register (PWPR). Modify the register after releasing the protection.
The ISEL bit to which IRQn is not specified is reserved. The ASEL bit to which analog input/output is not specified is reserved.

## 19.2.3 P1n Pin Function Control Registers (P1nPFS) (n = 4 to 7)

Address(es): P14PFS: 0008 C14Ch, P15PFS: 0008 C14Dh, P16PFS: 0008 C14Eh, P17PFS: 0008 C14Fh




| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b4 to b0 | PSEL[4:0] | Pin Function Select | These bits select the peripheral function. For individual pin functions, refer to the table below. | R/W |
| b5 | — | Reserved | This bit is read as 0. The write value should be 0. | R/W |
| b6 | ISEL | Interrupt Input Function Select | 0: Not used as IRQn input pin<br>1: Used as IRQn input pin<br>P14: IRQ4 (64 pins, 48 pins, 40 pins, 36 pins)<br>P15: IRQ5 (64 pins, 48 pins, 40 pins, 36 pins)<br>P16: IRQ6 (64 pins, 48 pins, 40 pins, 36 pins)<br>P17: IRQ7 (64 pins, 48 pins, 40 pins, 36 pins) | R/W |
| b7 | — | Reserved | This bit is read as 0. The write value should be 0. | R/W |

**Table 19.2 Register Settings for Input/Output Pin Functions in 64-Pin and 48-Pin**

| PSEL[4:0] Settings | Pin | | | |
|---|---|---|---|---|
| | P14 | P15 | P16 | P17 |
| 00000b (initial value) | Hi-Z | | | |
| 00001b | MTIOC3A | MTIOC0B | MTIOC3C | MTIOC3A |
| 00010b | MTCLKA | MTCLKB | MTIOC3D | MTIOC3B |
| 00011b | MTIOC0A | — | — | MTIOC0C |
| 00111b | — | — | RTCOUT | POE8# |
| 01001b | — | CLKOUT | ADTRG0# | — |
| 01010b | — | RXD1<br>SMISO1<br>SSCL1 | TXD1<br>SMOSI1<br>SSDA1 | SCK1 |
| 01011b | CTS1#<br>RTS1#<br>SS1# | — | — | — |
| 01100b | TXD12<br>SMOSI12<br>SSDA12<br>TXDX12<br>SIOX12 | — | — | RXD12<br>SMISO12<br>SSCL12<br>RXDX12 |
| 01101b | SSLA0 | RSPCKA | MOSIA | MISOA |
| 01111b | — | — | SCL0 | SDA0 |
| 10001b | — | — | USB0_VBUSEN | — |
| 10010b | — | — | USB0_VBUS*1 | — |
| 10011b | USB0_OVRCURA | — | USB0_OVRCURB | — |

—: Do not specify this value.

Note 1. 5 V tolerant.

**Table 19.3 Register Settings for Input/Output Pin Functions in 40-Pin and 36-Pin**

| PSEL[4:0] Settings | Pin | | | |
|---|---|---|---|---|
| | P14 | P15 | P16 | P17 |
| 00000b (initial value) | Hi-Z | | | |
| 00001b | MTIOC3A | MTIOC0B | MTIOC3C | MTIOC3A |
| 00010b | MTCLKA | MTCLKB | MTIOC3D | MTIOC3B |
| 00011b | MTIOC0A | — | — | MTIOC0C |
| 00111b | — | — | — | POE8# |
| 01001b | — | CLKOUT | ADTRG0# | — |
| 01010b | — | RXD1<br>SMISO1<br>SSCL1 | TXD1<br>SMOSI1<br>SSDA1 | SCK1 |
| 01011b | CTS1#<br>RTS1#<br>SS1# | — | — | — |
| 01100b | TXD12<br>SMOSI12<br>SSDA12<br>TXDX12<br>SIOX12 | — | — | RXD12<br>SMISO12<br>SSCL12<br>RXDX12 |
| 01101b | SSLA0 | RSPCKA | MOSIA | MISOA |
| 01111b | — | — | SCL0 | SDA0 |
| 10001b | — | — | USB0_VBUSEN | — |
| 10010b | — | — | USB0_VBUS*1 | — |
| 10011b | USB0_OVRCURA | — | USB0_OVRCURB | — |

—: Do not specify this value.

Note 1. 5 V tolerant.

## 19.2.4 P2n Pin Function Control Registers (P2nPFS) (n = 6 to 7)

Address(es): P26PFS: 0008 C156h, P27PFS: 0008 C157h

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | ASEL | ISEL | — | PSEL[4:0] | | | | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b4 to b0 | PSEL[4:0] | Pin Function Select | These bits select the peripheral function. For individual pin functions, refer to the table below. | R/W |
| b5 | — | Reserved | This bit is read as 0. The write value should be 0. | R/W |
| b6 | ISEL | Interrupt Input Function Select | 0: Not used as IRQn input pin<br>1: Used as IRQn input pin<br>P27: IRQ3 (64 pins, 48 pins, 40 pins, 36 pins) | R/W |
| b7 | ASEL | Analog Function Select | 0: Used as other than an analog pin<br>1: Used as an analog pin<br>P27: CMPA2 (64 pins, 48 pins, 40 pins, 36 pins) | R/W |

**Table 19.4 Register Settings for Input/Output Pin Functions in 64-Pin, 48-Pin, and 40-Pin**

| PSEL[4:0] Settings | Pin | |
|---|---|---|
| | P26 | P27 |
| 00000b (initial value) | Hi-Z | |
| 00001b | MTIOC2A | MTIOC2B |
| 00111b | — | CACREF |
| 01001b | — | ADTRG0# |
| 01010b | TXD1<br>SMOSI1<br>SSDA1 | SCK1 |
| 01100b | — | SCK12 |
| 10011b | USB0_VBUSEN | — |

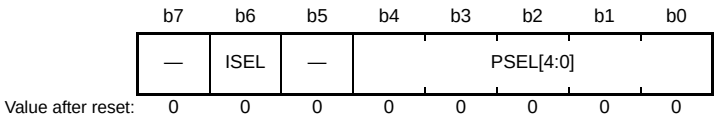
—: Do not specify this value.

**Table 19.5 Register Settings for Input/Output Pin Function in 36-Pin**

| PSEL[4:0] Settings | Pin |
|---|---|
| | P27 |
| 00000b (initial value) | Hi-Z |
| 00001b | MTIOC2B |
| 00111b | CACREF |
| 01001b | ADTRG0# |
| 01010b | SCK1 |
| 01100b | SCK12 |

## 19.2.5 P3n Pin Function Control Registers (P3nPFS) (n = 0 to 2, 5)

Address(es): P30PFS: 0008 C158h, P31PFS: 0008 C159h, P32PFS: 0008 C15Ah, P35PFS: 0008 C15Dh




| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b4 to b0 | PSEL[4:0] | Pin Function Select | These bits select the peripheral function. For individual pin functions, refer to the table below. | R/W |
| b5 | — | Reserved | This bit is read as 0. The write value should be 0. | R/W |
| b6 | ISEL | Interrupt Input Function Select | 0: Not used as IRQn and NMI input pin<br>1: Used as IRQn and NMI input pin<br>P30: IRQ0 (64 pins)<br>P31: IRQ1 (64 pins)<br>P32: IRQ2 (64 pins, 40 pins)<br>P35: NMI (64 pins, 48 pins, 40 pins, 36 pins) | R/W |
| b7 | — | Reserved | This bit is read as 0. The write value should be 0. | R/W |

**Table 19.6 Register Settings for Input/Output Pin Function in 64-Pin**

| PSEL[4:0] Settings | Pin | | |
|---|---|---|---|
| | P30 | P31 | P32 |
| 00000b (initial value) | Hi-Z | | |
| 00001b | MTIOC4B | MTIOC4D | MTIOC0C |
| 00111b | POE8# | — | RTCOUT |
| 01010b | RXD1<br>SMISO1<br>SSCL1 | — | — |
| 01011b | — | CTS1#<br>RTS1#<br>SS1# | — |

—: Do not specify this value.

**Table 19.7 Register Settings for Input/Output Pin Function in 40-Pin**

| PSEL[4:0] Settings | Pin |
|---|---|
| | P32 |
| 00000b (initial value) | Hi-Z |
| 00001b | MTIOC0C |

## 19.2.6 P4n Pin Function Control Registers (P4nPFS) (n = 0 to 4, 6)

Address(es): P40PFS: 0008 C160h, P41PFS: 0008 C161h, P42PFS: 0008 C162h, P43PFS: 0008 C163h, P44PFS: 0008 C164h, P46PFS: 0008 C166h

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | ASEL | — | — | — | — | — | — | — |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b6 to b0 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |
| b7 | ASEL | Analog Function Select | 0: Used as other than an analog pin<br>1: Used as an analog pin<br>P40: AN002 (64 pins, 48 pins)<br>P41: AN001 (64 pins, 48 pins, 40 pins, 36 pins)<br>P42: AN002 (64 pins, 48 pins, 40 pins, 36 pins)<br>P43: AN003 (64 pins)<br>P44: AN004 (64 pins)<br>P46: AN006 (64 pins, 48 pins, 40 pins) | R/W |

## 19.2.7 P5n Pin Function Select Registers (P5nPFS) (n = 4, 5)

Address(es): P54PFS: 0008 C16Ch, P55PFS: 0008 C16Dh

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | — | PSEL[4:0] | | | | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b4 to b0 | PSEL[4:0] | Pin Function Select | These bits select the peripheral function. For individual pin functions, refer to the table below. | R/W |
| b7 to b5 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |

**Table 19.8 Register Settings for Input/Output Pin Function in 64-Pin**

| PSEL[4:0] Settings | Pin | |
|---|---|---|
| | P54 | P55 |
| 00000b (initial value) | Hi-Z | |
| 00001b | MTIOC4B | MTIOC4D |

## 19.2.8 PAn Pin Function Control Registers (PAnPFS) (n = 0, 1, 3, 4, 6)

Address(es): PA0PFS: 0008 C190h, PA1PFS: 0008 C191h, PA3PFS: 0008 C193h, PA4PFS: 0008 C194h, PA6PFS: 0008 C196h

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | ISEL | — | PSEL[4:0] | | | | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b4 to b0 | PSEL[4:0] | Pin Function Select | These bits select the peripheral function. For individual pin functions, refer to the table below. | R/W |
| b5 | — | Reserved | This bit is read as 0. The write value should be 0. | R/W |
| b6 | ISEL | Interrupt Input Function Select | 0: Not used as IRQn input pin<br>1: Used as IRQn input pin<br>PA3: IRQ6 (64 pins, 48 pins, 40 pins, 36 pins)<br>PA4: IRQ5 (64 pins, 48 pins, 40 pins, 36 pins)<br>PA6: IRQ3 (64 pins, 48 pins, 40 pins, 36 pins) | R/W |
| b7 | — | Reserved | This bit is read as 0. The write value should be 0. | R/W |

**Table 19.9 Register Settings for Input/Output Pin Function in 64-Pin**

| PSEL[4:0] Settings | Pin | | | | |
|---|---|---|---|---|---|
| | PA0 | PA1 | PA3 | PA4 | PA6 |
| 00000b (initial value) | Hi-Z | | | | |
| 00001b | MTIOC4A | MTIOC0B | MTIOC0D | MTIC5U | MTIC5V |
| 00010b | — | MTCLKC | MTCLKD | MTCLKA | MTCLKB |
| 00011b | — | — | MTIOC1B | MTIOC2B | MTIOC2A |
| 00111b | CACREF | RTCOUT | POE0# | — | POE2# |
| 01010b | — | SCK5 | RXD5<br>SMISO5<br>SSCL5 | TXD5<br>SMOSI5<br>SSDA5 | — |
| 01011b | — | — | — | — | CTS5#<br>RTS5#<br>SS5# |
| 01101b | SSLA1 | SSLA2 | MISOA | SSLA0 | MOSIA |
| 01111b | — | — | — | — | SDA0 |

—: Do not specify this value.

**Table 19.10 Register Settings for Input/Output Pin Function in 48-Pin**

| PSEL[4:0] Settings | Pin | | | |
|---|---|---|---|---|
| | **PA1** | **PA3** | **PA4** | **PA6** |
| 00000b (initial value) | Hi-Z | | | |
| 00001b | MTIOC0B | MTIOC0D | MTIC5U | MTIC5V |
| 00010b | MTCLKC | MTCLKD | MTCLKA | MTCLKB |
| 00011b | — | MTIOC1B | MTIOC2B | MTIOC2A |
| 00111b | RTCOUT | POE0# | — | POE2# |
| 01010b | SCK5 | RXD5<br>SMISO5<br>SSCL5 | TXD5<br>SMOSI5<br>SSDA5 | — |
| 01011b | — | — | — | CTS5#<br>RTS5#<br>SS5# |
| 01101b | SSLA2 | MISOA | SSLA0 | MOSIA |
| 01111b | — | — | — | SDA0 |

—: Do not specify this value.

**Table 19.11 Register Settings for Input/Output Pin Function in 40-Pin**

| PSEL[4:0] Settings | Pin | | | |
|---|---|---|---|---|
| | **PA1** | **PA3** | **PA4** | **PA6** |
| 00000b (initial value) | Hi-Z | | | |
| 00001b | MTIOC0B | MTIOC0D | MTIC5U | MTIC5V |
| 00010b | MTCLKC | MTCLKD | MTCLKA | MTCLKB |
| 00011b | — | MTIOC1B | MTIOC2B | MTIOC2A |
| 00111b | — | POE0# | — | POE2# |
| 01010b | SCK5 | RXD5<br>SMISO5<br>SSCL5 | TXD5<br>SMOSI5<br>SSDA5 | — |
| 01011b | — | — | — | CTS5#<br>RTS5#<br>SS5# |
| 01101b | SSLA2 | MISOA | SSLA0 | MOSIA |
| 01111b | — | — | — | SDA0 |

—: Do not specify this value.

**Table 19.12 Register Settings for Input/Output Pin Function in 36-Pin**

| PSEL[4:0] Settings | Pin | | |
|---|---|---|---|
| | PA3 | PA4 | PA6 |
| 00000b (initial value) | Hi-Z | | |
| 00001b | MTIOC0D | MTIC5U | MTIC5V |
| 00010b | MTCLKD | MTCLKA | MTCLKB |
| 00011b | MTIOC1B | MTIOC2B | MTIOC2A |
| 00111b | POE0# | — | POE2# |
| 01010b | RXD5<br>SMISO5<br>SSCL5 | TXD5<br>SMOSI5<br>SSDA5 | — |
| 01011b | — | — | CTS5#<br>RTS5#<br>SS5# |
| 01101b | MISOA | SSLA0 | MOSIA |
| 01111b | — | — | SDA0 |

—: Do not specify this value.

## 19.2.9 PBn Pin Function Control Registers (PBnPFS) (n = 0, 1, 3, 5 to 7)

Address(es): PB0PFS: 0008 C198h, PB1PFS: 0008 C199h, PB3PFS: 0008 C19Bh, PB5PFS: 0008 C19Dh, PB6PFS: 0008 C19Eh, PB7PFS: 0008 C19Fh

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | ISEL | — | PSEL[4:0] | | | | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b4 to b0 | PSEL[4:0] | Pin Input/Output Function | These bits select the peripheral function. For individual pin functions, refer to the table below. | R/W |
| b5 | — | Reserved | This bit is read as 0. The write value should be 0. | R/W |
| b6 | ISEL | Interrupt Input Function Select | 0: Not used as IRQn input pin<br>1: Used as IRQn input pin<br>PB0: IRQ2 (64 pins, 48 pins, 40 pins, 36 pins)<br>PB1: IRQ4 (64 pins, 48 pins) | R/W |
| b7 | — | Reserved | This bit is read as 0. The write value should be 0. | R/W |

**Table 19.13 Register Settings for Input/Output Pin Function in 64-Pin**

| PSEL[4:0] Settings | Pin | | | | | |
|---|---|---|---|---|---|---|
| | PB0 | PB1 | PB3 | PB5 | PB6 | PB7 |
| 00000b (initial value) | Hi-Z | | | | | |
| 00001b | MTIC5W | MTIOC0C | MTIOC0A | MTIOC2A | MTIOC3D | MTIOC3B |
| 00010b | MTIOC0C | MTIOC4C | MTIOC4A | MTIOC1B | — | — |
| 00011b | — | — | MTIOC3B | — | — | — |
| 00111b | RTCOUT | — | POE3# | POE1# | — | — |
| 01001b | ADTRG0# | — | — | — | — | — |
| 01101b | RSPCKA | — | — | — | — | — |
| 01111b | SCL0 | — | — | — | — | — |
| 10011b | — | — | USB0_OVRCURA | — | — | — |

—: Do not specify this value.

**Table 19.14 Register Settings for Input/Output Pin Function in 48-Pin**

| PSEL[4:0] Settings | Pin | | | |
|---|---|---|---|---|
| | PB0 | PB1 | PB3 | PB5 |
| 00000b (initial value) | Hi-Z | | | |
| 00001b | MTIC5W | MTIOC0C | MTIOC0A | MTIOC2A |
| 00010b | MTIOC0C | MTIOC4C | MTIOC4A | MTIOC1B |
| 00011b | — | — | MTIOC3B | — |
| 00111b | RTCOUT | — | POE3# | POE1# |
| 01001b | ADTRG0# | — | — | — |
| 01101b | RSPCKA | — | — | — |
| 01111b | SCL0 | — | — | — |
| 10011b | — | — | USB0_OVRCURA | — |

—: Do not specify this value.

**Table 19.15 Register Settings for Input/Output Pin Function in 40-Pin and 36-Pin**

| PSEL[4:0] Settings | Pin | |
|---|---|---|
| | PB0 | PB3 |
| 00000b (initial value) | Hi-Z | |
| 00001b | MTIC5W | MTIOC0A |
| 00010b | MTIOC0C | MTIOC4A |
| 00011b | — | MTIOC3B |
| 00111b | — | POE3# |
| 01001b | ADTRG0# | — |
| 01101b | RSPCKA | — |
| 01111b | SCL0 | — |
| 10011b | — | USB0_OVRCURA |

—: Do not specify this value.

## 19.2.10 PCn Pin Function Control Registers (PCnPFS) (n = 2 to 7)

Address(es): PC2PFS: 0008 C1A2h, PC3PFS: 0008 C1A3h, PC4PFS: 0008 C1A4h, PC5PFS: 0008 C1A5h, PC6PFS: 0008 C1A6h, PC7PFS: 0008 C1A7h

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | ISEL | — | PSEL[4:0] | | | | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b4 to b0 | PSEL[4:0] | Pin Input/Output Function | These bits select the peripheral function. For individual pin functions, refer to the table below. | R/W |
| b5 | — | Reserved | This bit is read as 0. The write value should be 0. | R/W |
| b6 | ISEL | Interrupt Input Function Select | 0: Not used as IRQn input pin<br>1: Used as IRQn input pin<br>PC4: IRQ2 (64 pins, 48 pins, 40 pins, 36 pins) | R/W |
| b7 | — | Reserved | This bit is read as 0. The write value should be 0. | R/W |

**Table 19.16 Register Settings for Input/Output Pin Function in 64-Pin**

| PSEL[4:0] Settings | Pin | | | | | |
|---|---|---|---|---|---|---|
| | PC2 | PC3 | PC4 | PC5 | PC6 | PC7 |
| 00000b (initial value) | Hi-Z | | | | | |
| 00001b | MTIOC4B | MTIOC4D | MTIOC3D | MTIOC3B | MTIOC3C | MTIOC3A |
| 00010b | — | — | MTCLKC | MTCLKD | MTCLKA | MTCLKB |
| 00111b | — | — | POE0# | — | — | CACREF |
| 01001b | — | — | CLKOUT | — | — | — |
| 01010b | RXD5<br>SMISO5<br>SSCL5 | TXD5<br>SMOSI5<br>SSDA5 | SCK5 | — | — | — |
| 01011b | — | — | — | SCK1 | RXD1<br>SMISO1<br>SSCL1 | TXD1<br>SMOSI1<br>SSDA1 |
| 01101b | SSLA3 | — | SSLA0 | RSPCKA | MOSIA | MISOA |
| 10001b | — | — | USB0_VBUSEN | — | — | — |
| 10010b | — | — | USB0_VBUS*1 | — | — | — |
| 10011b | — | — | — | USB0_ ID | USB0_EXICEN | USB0_OVRCURB |

—: Do not specify this value.
Note 1. Not 5 V tolerant.

**Table 19.17 Register Settings for Input/Output Pin Function in 48-Pin**

| PSEL[4:0] Settings | Pin | | | |
|---|---|---|---|---|
| | PC4 | PC5 | PC6 | PC7 |
| 00000b (initial value) | Hi-Z | | | |
| 00001b | MTIOC3D | MTIOC3B | MTIOC3C | MTIOC3A |
| 00010b | MTCLKC | MTCLKD | MTCLKA | MTCLKB |
| 00111b | POE0# | — | — | CACREF |
| 01001b | CLKOUT | — | — | — |
| 01010b | SCK5 | — | — | — |
| 01011b | — | SCK1 | RXD1<br>SMISO1<br>SSCL1 | TXD1<br>SMOSI1<br>SSDA1 |
| 01101b | SSLA0 | RSPCKA | MOSIA | MISOA |
| 10001b | USB0_VBUSEN | — | — | — |
| 10010b | USB0_VBUS*1 | — | — | — |
| 10011b | — | USB0_ ID | USB0_EXICEN | USB0_OVRCURB |

—: Do not specify this value.
Note 1. Not 5 V tolerant.

**Table 19.18 Register Settings for Input/Output Pin Function in 40-Pin and 36-Pin**

| PSEL[4:0] Settings | Pin |
|---|---|
| | PC4 |
| 00000b (initial value) | Hi-Z |
| 00001b | MTIOC3D |
| 00010b | MTCLKC |
| 00111b | POE0# |
| 01001b | CLKOUT |
| 01010b | SCK5 |
| 01101b | SSLA0 |
| 10001b | USB0_VBUSEN |
| 10010b | USB0_VBUS*1 |

—: Do not specify this value.
Note 1. Not 5 V tolerant.

## 19.2.11 PEn Pin Function Control Registers (PEnPFS) (n = 0 to 7)

Address(es): PE0PFS: 0008 C1B0h, PE1PFS: 0008 C1B1h, PE2PFS: 0008 C1B2h, PE3PFS: 0008 C1B3h, PE4PFS: 0008 C1B4h, PE5PFS: 0008 C1B5h, PE6PFS: 0008 C1B6h, PE7PFS: 0008 C1B7h

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | ASEL | ISEL | — | PSEL[4:0] | | | | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b4 to b0 | PSEL[4:0] | PEn Pin Input/Output Function | These bits select the peripheral function. For individual pin functions, refer to the table below. | R/W |
| b5 | — | Reserved | This bit is read as 0. The write value should be 0. | R/W |
| b6 | ISEL | Interrupt Input Function Select | 0: Not used as IRQn input pin<br>1: Used as IRQn input pin<br>PE0: IRQ0 (64 pins, 48 pins, 40 pins, 36 pins)<br>PE1: IRQ1 (64 pins, 48 pins, 40 pins, 36 pins)<br>PE2: IRQ7 (64 pins, 48 pins, 40 pins, 36 pins)<br>PE3: IRQ3 (64 pins, 48 pins, 40 pins, 36 pins)<br>PE4: IRQ4 (64 pins, 48 pins, 40 pins, 36 pins)<br>PE5: IRQ5 (64 pins)<br>PE6: IRQ6 (64 pins)<br>PE7: IRQ7 (64 pins, 48 pins) | R/W |
| b7 | ASEL | PEn Analog Function Select | 0: Used as other than an analog pin<br>1: Used as an analog pin<br>PE0: AN008 (64 pins, 48 pins, 40 pins, 36 pins)<br>PE1: AN009 (64 pins, 48 pins, 40 pins, 36 pins)<br>PE2: AN010 (64 pins, 48 pins, 40 pins, 36 pins)<br>PE3: AN011 (64 pins, 48 pins, 40 pins, 36 pins)<br>PE4: AN012 (64 pins, 48 pins, 40 pins, 36 pins)<br>PE5: AN013 (64 pins)<br>PE6: AN014 (64 pins)<br>PE7: AN015 (64 pins, 48 pins) | R/W |

**Table 19.19 Register Settings for Input/Output Pin Function in 64-Pin**

| PSEL[4:0] Settings | Pin PE0 | PE1 | PE2 | PE3 | PE4 | PE5 | PE6 | PE7 |
|---|---|---|---|---|---|---|---|---|
| 00000b (initial value) | Hi-Z | | | | | | | |
| 00001b | — | MTIOC4C | MTIOC4A | MTIOC4B | MTIOC4D | MTIOC4C | — | — |
| 00010b | MTIOC2A | — | — | MTIOC1B | MTIOC1A | MTIOC2B | — | — |
| 00011b | — | — | — | MTIOC0A | MTIOC3A | — | — | — |
| 00111b | POE3# | — | — | POE8# | — | — | — | — |
| 01100b | SCK12 | TXD12<br>SMOSI12<br>SSDA12<br>TXDX12<br>SIOX12 | RXD12<br>SMISO12<br>SSCL12<br>RXDX12 | CTS12#<br>RTS12#<br>SS12# | — | — | — | — |
| 01101b | — | — | — | RSPCKA | MOSIA | — | — | — |

—: Do not specify this value.

**Table 19.20 Register Settings for Input/Output Pin Function in 48-Pin**

| PSEL[4:0] Settings | Pin PE0 | PE1 | PE2 | PE3 | PE4 | PE7 |
|---|---|---|---|---|---|---|
| 00000b (initial value) | Hi-Z | | | | | |
| 00001b | — | MTIOC4C | MTIOC4A | MTIOC4B | MTIOC4D | — |
| 00010b | MTIOC2A | — | — | MTIOC1B | MTIOC1A | — |
| 00011b | — | — | — | MTIOC0A | MTIOC3A | — |
| 00111b | POE3# | — | — | POE8# | — | — |
| 01100b | SCK12 | TXD12<br>SMOSI12<br>SSDA12<br>TXDX12<br>SIOX12 | RXD12<br>SMISO12<br>SSCL12<br>RXDX12 | CTS12#<br>RTS12#<br>SS12# | — | — |
| 01101b | — | — | — | RSPCKA | MOSIA | — |

—: Do not specify this value.

**Table 19.21 Register Settings for Input/Output Pin Function in 36-Pin and 40-Pin**

| PSEL[4:0] Settings | Pin PE0 | PE1 | PE2 | PE3 | PE4 |
|---|---|---|---|---|---|
| 00000b (initial value) | Hi-Z | | | | |
| 00001b | — | MTIOC4C | MTIOC4A | MTIOC4B | MTIOC4D |
| 00010b | MTIOC2A | — | — | MTIOC1B | MTIOC1A |
| 00011b | — | — | — | MTIOC0A | MTIOC3A |
| 00111b | POE3# | — | — | POE8# | — |
| 01100b | SCK12 | TXD12<br>SMOSI12<br>SSDA12<br>TXDX12<br>SIOX12 | RXD12<br>SMISO12<br>SSCL12<br>RXDX12 | CTS12#<br>RTS12#<br>SS12# | — |
| 01101b | — | — | — | RSPCKA | MOSIA |

—: Do not specify this value.

## 19.2.12 PJn Pin Function Control Registers (PJnPFS) (n = 6, 7)

Address(es): PJ6PFS: 0008 C1D6h, PJ7PFS: 0008 C1D7h

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | ASEL | — | — | — | — | — | — | — |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b6 to b0 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |
| b7 | ASEL | Analog Function Select | PJ6PFS.ASEL bit (64-pin, 48-pin, 40-pin, 36-pin)<br>0: The AVCC0 pin is selected as the reference power supply pin for high-electric potential<br>1: The VREFH0 pin is selected as the reference power supply pin for low-electric potential.<br>PJ7PFS.ASEL bit (64-pin, 48-pin, 40-pin, 36-pin)<br>0: The AVSS0 pin is selected as the reference power supply ground pin for low-electric potential.<br>1: The VREFL0 pin is selected as the reference power supply ground pin for high-electric potential. | R/W |

## 19.3 Usage Notes

### 19.3.1 Procedure for Specifying Input/Output Pin Functions

Use the following procedure to specify the input/output pin functions.

1. Clear the port mode register (PMR) to 0 to select the general I/O port function.
2. Set the I/O register in the peripheral module to set the I/O signal assigned to the target pin.
3. Set the write-protect register (PWPR) to enable writing to the Pmn pin function control register (PmnPFS). (m = 0 to 5, A to C, E, J, n = 0 to 7)
4. Specify the input/output function for the pin through the PSEL[4:0] bit settings in the PmnPFS register.
5. Clear the PWPR.PFSWE bit to 0 to disable writing to the PmnPFS register.
6. Set the PMR register to 1 as necessary to switch to the selected input/output function for the pin.

### 19.3.2 Notes on MPC Register Setting

1. Only set the Pmn pin function control register (PmnPFS) while the PMR register for the target pin is cleared to 0. If the PmnPFS is set while the PMR register is 1, unexpected edges may be input through the input pin or unexpected pulses are output through the output pin.
2. Only the allowed values (functions) should be specified in the PmnPFS register. If a value that is not allowed for the register is specified, correct operation is not guaranteed.
3. Do not assign a single function to multiple pins through the MPC register settings.
4. Analog input functions for the A/D converter are multiplexed with pins of ports 4 and E. If a pin is to be used as an analog input, avoid loss of resolution by setting the given bits of the port mode register (PMR) and of the port direction register (PDR) to 0, i.e. configuring the pin as a general input, and setting the PmnPFS.ASEL bit to 1.

5. Points to note regarding the port mode register (PMR), port direction register (PDR), and Pmn pin function control register (PmnPFS) settings for pins that have multiplexed pin functions are listed in Table 19.22.

**Table 19.22 Register Settings**

| Item | PMR.Bn | PDR.Bn | PmnPFS | | | Point to Note |
|---|---|---|---|---|---|---|
| | | | ASEL | ISEL | PSEL[4:0] | |
| After a reset | 0 | 0 | 0 | 0 | 00000b | Pins function as general input port pins after release from the reset state. |
| General input ports | 0 | 0 | 0 | 0/1 | × | Set the ISEL bit to 1 if these are multiplexed with interrupt inputs. |
| General output ports | 0 | 1 | 0 | 0 | × | |
| Peripheral functions | 1 | X | 0 | 0/1 | Peripheral functions (see Table 19.2 to Table 19.21) | Set the ISEL bit to 1 if these are multiplexed with interrupt inputs. |
| Interrupt inputs | 0 | 0 | 0 | 1 | × | |
| NMI | × | × | × | ×*1 | × | Register settings are not required. |
| Analog inputs and outputs | 0 | 0 | 1 | ×*1 | × | Set these as general input port pins so that the output buffers are turned off. |
| XCIN | 0 | 0 | × | ×*1 | × | Set these as general input port pins so that the output buffers are turned off. |

×: Setting not required.
0/1: Setting the PmnPFS.ISEL bit to 0 makes the pin incapable of functioning as an IRQ pin.
Setting the PmnPFS.ISEL bit to 1 makes the pin capable of functioning as an IRQ pin (if the IRQ is selected from the multiplexed functions).
Note 1. The pin does not function as the IRQn input pin even if the PmnPFS.ISEL bit is set to 1.

Note: • The pin state is readable when the PmnPFS.ASEL bit is 0.
• If the value of the PmnPFS.PSEL[4:0] bits is to be changed, do so while the PMR.Bn bit is 0.
• If an RIIC function is assigned to a port pin, clear the PCR.Bn (to 0); pulling up is automatically turned off for outputs from peripheral modules other than the RIIC.

### 19.3.3 Note on Using Analog Functions

When an analog function is in use, configure the pin as a general input by setting the given bits of the port mode register (PMR) and of the port direction register (PDR) to 0, and then set the ASEL bit in the port mn pin function select register (PmnPFS) to 1.

# 20. Multi-Function Timer Pulse Unit 2 (MTU2a)

## 20.1 Overview

This MCU has an on-chip multi-function timer pulse unit 2 (MTU). Each unit comprises a 16-bit timer with six channels (MTU0 to MTU5).

Table 20.1 lists the specifications of the MTU, and Table 20.2 lists the function list. Figure 20.1 shows a block diagram of the MTU.

**Table 20.1 MTU Specifications**

| Item | Description |
|---|---|
| Pulse input/output | 16 lines max. |
| Pulse input | 3 lines |
| Count clock | Eight clocks or seven clocks for each channel (four clocks for MTU5) |
| Available operations | [MTU0 to MTU4]<br>• Waveform output at compare match<br>• Input capture function (noise filter set function)<br>• Counter clear operation<br>• Simultaneous writing to multiple timer counters (TCNT)<br>• Simultaneous clearing by compare match or input capture<br>• Simultaneous register input/output by synchronous counter operation<br>• A maximum of 12-phase PWM output is available in combination with synchronous operation |
| | [MTU0, MTU3, MTU4]<br>• Buffer operation specifiable<br>• AC synchronous motor (brushless DC motor) drive mode using complementary PWM output and reset-synchronized PWM output is settable and the selection of two types of waveform outputs (chopping and level) is possible. |
| | [MTU1, MTU2]<br>• Phase counting mode specifiable independently<br>• Cascade connection operation |
| | [MTU3, MTU4]<br>• A total of six-phase waveform output, which includes three phases each for positive and negative complementary PWM or reset PWM output, by interlocking operation |
| | [MTU5]<br>• Dead time compensation counter<br>• Input capture function (noise filter set function)<br>• Counter clear operation |
| Complementary PWM mode | • Interrupts at the crest and trough of the counter value<br>• A/D converter start triggers can be skipped |
| Interrupt sources | 28 sources |
| Buffer operation | Automatic transfer of register data |
| Trigger generation | A/D converter start trigger can be generated |
| Low power consumption function | Module stop state can be set |

**Table 20.2 MTU Functions (1/2)**

| Item | | MTU0 | MTU1 | MTU2 | MTU3 | MTU4 | MTU5 |
|---|---|---|---|---|---|---|---|
| Count clock | | PCLK/1<br>PCLK/4<br>PCLK/16<br>PCLK/64<br>MTCLKA<br>MTCLKB<br>MTCLKC<br>MTCLKD | PCLK/1<br>PCLK/4<br>PCLK/16<br>PCLK/64<br>PCLK/256<br>MTCLKA<br>MTCLKB | PCLK/1<br>PCLK/4<br>PCLK/16<br>PCLK/64<br>PCLK/1024<br>MTCLKA<br>MTCLKB<br>MTCLKC | PCLK/1<br>PCLK/4<br>PCLK/16<br>PCLK/64<br>PCLK/256<br>PCLK/1024<br>MTCLKA<br>MTCLKB | PCLK/1<br>PCLK/4<br>PCLK/16<br>PCLK/64<br>PCLK/256<br>PCLK/1024<br>MTCLKA<br>MTCLKB | PCLK/1<br>PCLK/4<br>PCLK/16<br>PCLK/64 |
| General registers (TGR) | | TGRA<br>TGRB<br>TGRE | TGRA<br>TGRB | TGRA<br>TGRB | TGRA<br>TGRB | TGRA<br>TGRB | TGRU<br>TGRV<br>TGRW |
| General registers/ buffer registers | | TGRC<br>TGRD<br>TGRF | — | — | TGRC<br>TGRD | TGRC<br>TGRD | — |
| I/O pins | | MTIOC0A<br>MTIOC0B<br>MTIOC0C<br>MTIOC0D | MTIOC1A<br>MTIOC1B | MTIOC2A<br>MTIOC2B | MTIOC3A<br>MTIOC3B<br>MTIOC3C<br>MTIOC3D | MTIOC4A<br>MTIOC4B<br>MTIOC4C<br>MTIOC4D | Input pins<br>MTIC5U<br>MTIC5V<br>MTIC5W |
| Counter clear function | | TGR compare match or input capture | TGR compare match or input capture | TGR compare match or input capture | TGR compare match or input capture | TGR compare match or input capture | TGR compare match or input capture |
| Compare match output | Low output | ○ | ○ | ○ | ○ | ○ | — |
| | High output | ○ | ○ | ○ | ○ | ○ | — |
| | Toggle output | ○ | ○ | ○ | ○ | ○ | — |
| Input capture function | | ○ | ○ | ○ | ○ | ○ | ○ |
| Synchronous operation | | ○ | ○ | ○ | ○ | ○ | — |
| PWM mode 1 | | ○ | ○ | ○ | ○ | ○ | — |
| PWM mode 2 | | ○ | ○ | ○ | — | — | — |
| Complementary PWM mode | | — | — | — | ○ | ○ | — |
| Reset-synchronized PWM | | — | — | — | ○ | ○ | — |
| AC synchronous motor drive mode | | ○ | — | — | ○ | ○ | — |
| Phase counting mode | | — | ○ | ○ | — | — | — |
| Buffer operation | | ○ | — | — | ○ | ○ | — |
| Dead time compensation counter function | | — | — | — | — | — | ○ |
| DTC activation | | TGR compare match or input capture | TGR compare match or input capture | TGR compare match or input capture | TGR compare match or input capture | TGR compare match or input capture and TCNT overflow or underflow | TGR compare match or input capture |
| A/D converter start trigger | | TGRA compare match or input capture<br>TGRB compare match or input capture<br>TGRE compare match<br>TGRF compare match | TGRA compare match or input capture | TGRA compare match or input capture | TGRA compare match or input capture | TGRA compare match or input capture<br>TCNT underflow (trough) in complementary PWM mode | — |

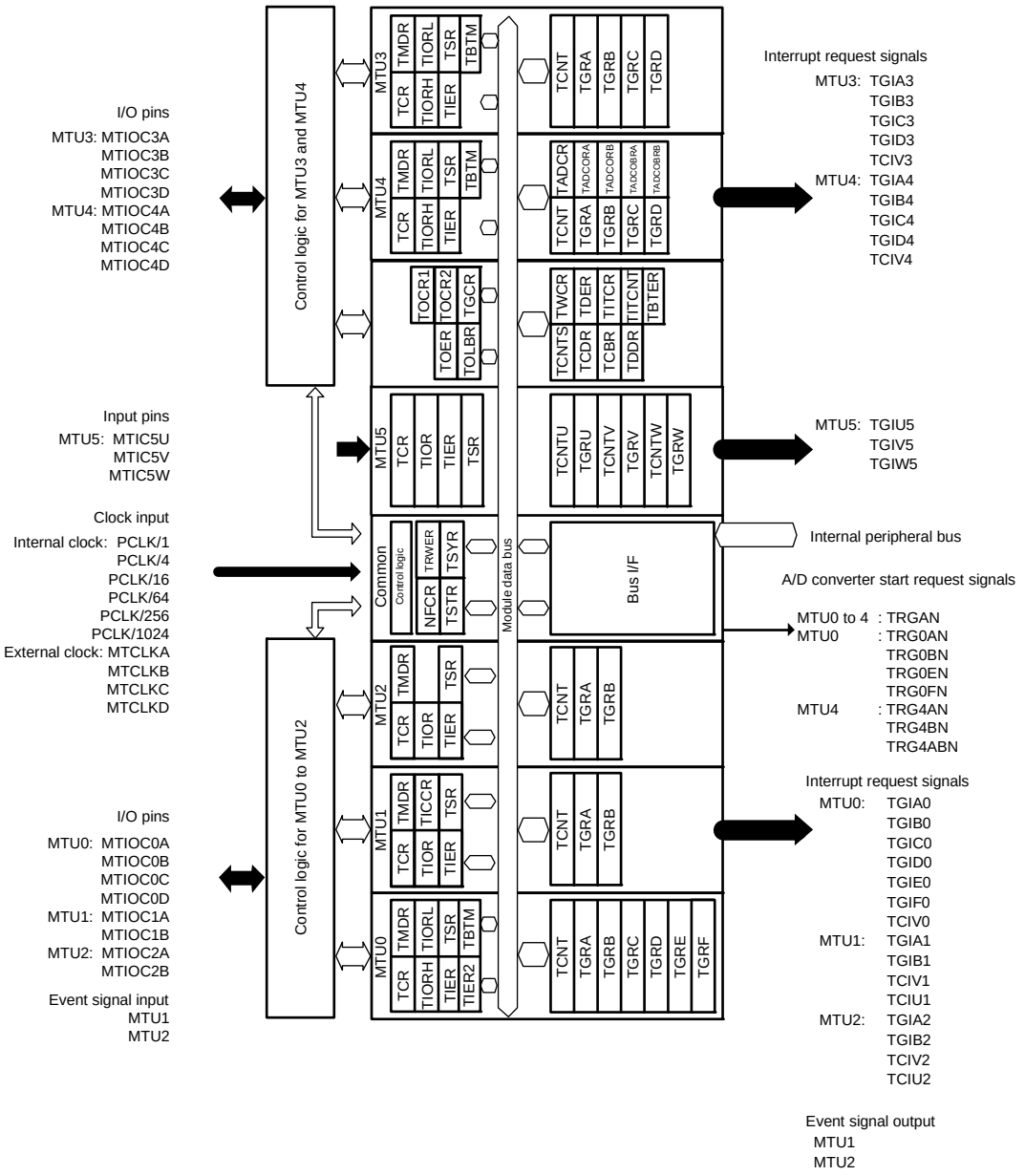
**Table 20.2 MTU Functions (2/2)**

| Item | MTU0 | MTU1 | MTU2 | MTU3 | MTU4 | MTU5 |
|---|---|---|---|---|---|---|
| Interrupt sources | 7 sources<br>• Compare match or input capture 0A<br>• Compare match or input capture 0B<br>• Compare match or input capture 0C<br>• Compare match or input capture 0D<br>• Compare match 0E<br>• Compare match 0F<br>• Overflow | 4 sources<br>• Compare match or input capture 1A<br>• Compare match or input capture 1B<br>• Overflow<br>• Underflow | 4 sources<br>• Compare match or input capture 2A<br>• Compare match or input capture 2B<br>• Overflow<br>• Underflow | 5 sources<br>• Compare match or input capture 3A<br>• Compare match or input capture 3B<br>• Compare match or input capture 3C<br>• Compare match or input capture 3D<br>• Overflow | 5 sources<br>• Compare match or input capture 4A<br>• Compare match or input capture 4B<br>• Compare match or input capture 4C<br>• Compare match or input capture 4D<br>• Overflow or underflow | 3 sources<br>• Compare match or input capture 5U<br>• Compare match or input capture 5V<br>• Compare match or input capture 5W |
| Event link function (output) | — | 4 sources<br>• Compare match 1A<br>• Compare match 1B<br>• Overflow<br>• Underflow | 4 sources<br>• Compare match 2A<br>• Compare match 2B<br>• Overflow<br>• Underflow | 6 sources<br>• Compare match 3A<br>• Compare match 3B<br>• Compare match 3C<br>• Compare match 3D<br>• Overflow<br>• Underflow | 6 sources<br>• Compare match 4A<br>• Compare match 4B<br>• Compare match 4C<br>• Compare match 4D<br>• Overflow<br>• Underflow | — |
| Event link function (input) | — | (1) Count start operation<br>(2) Input capture operation (TRGA capture)<br>(3) Count restart operation | (1) Count start operation<br>(2) Input capture operation (TRGA capture)<br>(3) Count restart operation | (1) Count start operation<br>(2) Input capture operation (TRGA capture)<br>(3) Count restart operation | (1) Count start operation<br>(2) Input capture operation (TRGA capture)<br>(3) Count restart operation | — |
| A/D converter start request delaying function | — | — | — | — | • A/D converter start request at a match between TADCORA and TCNT or A/D converter start request at a match between TADCORB and TCNT | — |
| Interrupt skipping function | — | — | — | • Skips TGRA compare match interrupts | • Skips TCIV interrupts | — |
| Module stop function | MSTPCRA.MSTPA9[*1] | | | | | |

○: Possible

—: Not possible

Note 1. For details on the module stop function, refer to section 11, Low Power Consumption.




| | |
|---|---|
| TSTR: | Timer start register |
| TSYR: | Timer synchronous register |
| TCR: | Timer control register |
| NFCR: | Noise filter control register |
| TMDR: | Timer mode register |
| TIOR: | Timer I/O control register |
| TIORH: | Timer I/O control register H |
| TIORL: | Timer I/O control register L |
| TIER: | Timer interrupt enable register |
| TIER2: | Timer interrupt enable register 2 |
| TGCR: | Timer gate control register |
| TOER: | Timer output master enable register |
| TOCR1: | Timer output control register 1 |
| TOCR2: | Timer output control register 2 |
| TSR: | Timer status register |
| TBTM: | Timer buffer operation transfer mode register |
| TCNT, TCNTU, TCNTV, TCNTW: | Timer counter |
| TCNTS: | Timer subcounter |
| TRWER: | Timer read/write enable register |
| TOLBR: | Timer output level buffer register |
| TCDR: | Timer cycle data register |
| TCBR: | Timer cycle buffer register |
| TDDR: | Timer dead time data register |
| TGRA: | Timer general register A |
| TGRB: | Timer general register B |
| TGRC: | Timer general register C |
| TGRD: | Timer general register D |
| TGRE: | Timer general register E |
| TGRF: | Timer general register F |
| TGRU: | Timer general register U |
| TGRV: | Timer general register V |
| TGRW: | Timer general register W |
| TICCR: | Timer input capture control register |
| TWCR: | Timer waveform control register |
| TADCR: | Timer A/D converter start request control register |
| TADCORA: | Timer A/D converter start request cycle set register A |
| TADCORB: | Timer A/D converter start request cycle set register B |
| TADCOBRA: | Timer A/D converter start request cycle set buffer register A |
| TADCOBRB: | Timer A/D converter start request cycle set buffer register B |
| TITCR | :Timer interrupt skipping set registers |
| TITCNT: | Timer interrupt skipping counters register |
| TBTER: | Timer buffer transfer set register |
| TDER: | Timer dead time enable register |

**Figure 20.1 MTU Block Diagram**

Table 20.3 lists the I/O pins to be used by the MTU.

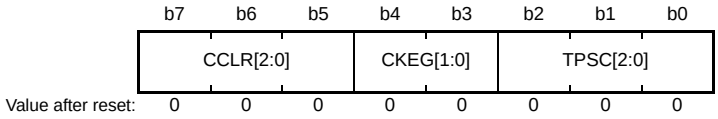
**Table 20.3 MTU I/O Pins**

| Module Symbol | Pin Name | I/O | Function |
|---|---|---|---|
| MTU | MTCLKA | Input | External clock A input pin (MTU1 phase counting mode A phase input) |
| | MTCLKB | Input | External clock B input pin (MTU1 phase counting mode B phase input) |
| | MTCLKC | Input | External clock C input pin (MTU2 phase counting mode A phase input) |
| | MTCLKD | Input | External clock D input pin (MTU2 phase counting mode B phase input) |
| MTU0 | MTIOC0A | I/O | TGRA0 input capture input/output compare output/PWM output pin |
| | MTIOC0B | I/O | TGRB0 input capture input/output compare output/PWM output pin |
| | MTIOC0C | I/O | TGRC0 input capture input/output compare output/PWM output pin |
| | MTIOC0D | I/O | TGRD0 input capture input/output compare output/PWM output pin |
| MTU1 | MTIOC1A | I/O | TGRA1 input capture input/output compare output/PWM output pin |
| | MTIOC1B | I/O | TGRB1 input capture input/output compare output/PWM output pin |
| MTU2 | MTIOC2A | I/O | TGRA2 input capture input/output compare output/PWM output pin |
| | MTIOC2B | I/O | TGRB2 input capture input/output compare output/PWM output pin |
| MTU3 | MTIOC3A | I/O | TGRA3 input capture input/output compare output/PWM output pin |
| | MTIOC3B | I/O | TGRB3 input capture input/output compare output/PWM output pin |
| | MTIOC3C | I/O | TGRC3 input capture input/output compare output/PWM output pin |
| | MTIOC3D | I/O | TGRD3 input capture input/output compare output/PWM output pin |
| MTU4 | MTIOC4A | I/O | TGRA4 input capture input/output compare output/PWM output pin |
| | MTIOC4B | I/O | TGRB4 input capture input/output compare output/PWM output pin |
| | MTIOC4C | I/O | TGRC4 input capture input/output compare output/PWM output pin |
| | MTIOC4D | I/O | TGRD4 input capture input/output compare output/PWM output pin |
| MTU5 | MTIC5U | Input | TGRU5 input capture input/external pulse input pin |
| | MTIC5V | Input | TGRV5 input capture input/external pulse input pin |
| | MTIC5W | Input | TGRW5 input capture input/external pulse input pin |

## 20.2 Register Descriptions

### 20.2.1 Timer Control Register (TCR)

Address(es): MTU0.TCR 0008 8700h, MTU1.TCR 0008 8780h, MTU2.TCR 0008 8800h, MTU3.TCR 0008 8600h, MTU4.TCR 0008 8601h, MTU5.TCRU 0008 8884h, MTU5.TCRV 0008 8894h, MTU5.TCRW 0008 88A4h




| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b2 to b0 | TPSC[2:0] | Time Prescaler Select | See Table 20.6 to Table 20.10. | R/W |
| b4, b3 | CKEG[1:0] | Clock Edge Select | b4 b3<br>0 0: Count at rising edge<br>0 1: Count at falling edge<br>1 x: Count at both edges | R/W |
| b7 to b5 | CCLR[2:0] | Counter Clear | See Table 20.4 and Table 20.5. | R/W |

x: Don't care

The MTU has a total of eight TCR registers, one each for MTU0 to MTU4 and three (TCRU, TCRV, and TCRW) for MTU5.
The TCR register controls the TCNT operation for each channel. TCR values should be specified only while TCNT operation is stopped.

**TPSC[2:0] Bits (Time Prescaler Select)**

These bits select the TCNT counter clock. The clock source can be selected for each channel. See Table 20.6 to Table 20.10 for details.

**CKEG[1:0] Bits (Clock Edge Select)**

These bits select the input clock edge. When the input clock is counted at both edges, the input clock period is halved (e.g. PCLK/4 clock at both edges = PCLK/2 clock at rising edge). If phase counting mode is used on MTU1 and MTU2, the setting of these bits is ignored and the phase counting mode setting has priority. Internal clock edge selection is valid when the input clock is PCLK/4 clock or slower. When PCLK/1 clock or the overflow/underflow in another channel is selected for the input clock, a value can be written to these bits but counter operation compiles with the initial value.

**CCLR[2:0] Bits (Counter Clear)**

These bits select the TCNT counter clearing source. See Table 20.4 and Table 20.5 for details.

**Table 20.4 CCLR[2:0] (MTU0, MTU3, and MTU4)**

| Channel | Bit 7 CCLR2 | Bit 6 CCLR1 | Bit 5 CCLR0 | Description |
|---|---|---|---|---|
| MTU0, MTU3, MTU4 | 0 | 0 | 0 | TCNT clearing disabled |
| | 0 | 0 | 1 | TCNT cleared by TGRA compare match/input capture |
| | 0 | 1 | 0 | TCNT cleared by TGRB compare match/input capture |
| | 0 | 1 | 1 | TCNT cleared by counter clearing in another channel performing synchronous clearing/synchronous operation*1 |
| | 1 | 0 | 0 | TCNT clearing disabled |
| | 1 | 0 | 1 | TCNT cleared by TGRC compare match/input capture*2 |
| | 1 | 1 | 0 | TCNT cleared by TGRD compare match/input capture*2 |
| | 1 | 1 | 1 | TCNT cleared by counter clearing in another channel performing synchronous clearing/synchronous operation*1 |

Note 1. Synchronous operation is selected by setting the TSYR.SYNCn bit (n = 0, 3, 4) to 1.
Note 2. When TGRC or TGRD is used as a buffer register, TCNT is not cleared because the buffer register setting has priority and compare match/input capture does not occur.

**Table 20.5 CCLR[2:0] (MTU1 and MTU2)**

| Channel | Bit 7 Reserved*2 | Bit 6 CCLR1 | Bit 5 CCLR0 | Description |
|---|---|---|---|---|
| MTU1, MTU2 | 0 | 0 | 0 | TCNT clearing disabled |
| | 0 | 0 | 1 | TCNT cleared by TGRA compare match/input capture |
| | 0 | 1 | 0 | TCNT cleared by TGRB compare match/input capture |
| | 0 | 1 | 1 | TCNT cleared by counter clearing in another channel performing synchronous clearing/synchronous operation*1 |

Note 1. Synchronous operation is selected by setting the TSYR.SYNCn bit (n = 1, 2) to 1.
Note 2. Bit 7 is reserved in MTU1 and MTU2. This bit is read as 0. The write value should be 0.

**Table 20.6 TPSC[2:0] (MTU0)**

| Channel | Bit 2 TPSC2 | Bit 1 TPSC1 | Bit 0 TPSC0 | Description |
|---|---|---|---|---|
| MTU0 | 0 | 0 | 0 | Internal clock: counts on PCLK/1 clock |
| | 0 | 0 | 1 | Internal clock: counts on PCLK/4 clock |
| | 0 | 1 | 0 | Internal clock: counts on PCLK/16 clock |
| | 0 | 1 | 1 | Internal clock: counts on PCLK/64 clock |
| | 1 | 0 | 0 | External clock: counts on MTCLKA pin input |
| | 1 | 0 | 1 | External clock: counts on MTCLKB pin input |
| | 1 | 1 | 0 | External clock: counts on MTCLKC pin input |
| | 1 | 1 | 1 | External clock: counts on MTCLKD pin input |

**Table 20.7 TPSC[2:0] (MTU1)**

| Channel | Bit 2 TPSC2 | Bit 1 TPSC1 | Bit 0 TPSC0 | Description |
|---|---|---|---|---|
| MTU1 | 0 | 0 | 0 | Internal clock: counts on PCLK/1 clock |
| | 0 | 0 | 1 | Internal clock: counts on PCLK/4 clock |
| | 0 | 1 | 0 | Internal clock: counts on PCLK/16 clock |
| | 0 | 1 | 1 | Internal clock: counts on PCLK/64 clock |
| | 1 | 0 | 0 | External clock: counts on MTCLKA pin input |
| | 1 | 0 | 1 | External clock: counts on MTCLKB pin input |
| | 1 | 1 | 0 | Internal clock: counts on PCLK/256 clock |
| | 1 | 1 | 1 | Counts on MTU2.TCNT overflow/underflow |

Note: • This setting is ignored when MTU1 is in phase counting mode.

**Table 20.8 TPSC[2:0] (MTU2)**

| Channel | Bit 2 TPSC2 | Bit 1 TPSC1 | Bit 0 TPSC0 | Description |
|---|---|---|---|---|
| MTU2 | 0 | 0 | 0 | Internal clock: counts on PCLK/1 clock |
| | 0 | 0 | 1 | Internal clock: counts on PCLK/4 clock |
| | 0 | 1 | 0 | Internal clock: counts on PCLK/16 clock |
| | 0 | 1 | 1 | Internal clock: counts on PCLK/64 clock |
| | 1 | 0 | 0 | External clock: counts on MTCLKA pin input |
| | 1 | 0 | 1 | External clock: counts on MTCLKB pin input |
| | 1 | 1 | 0 | External clock: counts on MTCLKC pin input |
| | 1 | 1 | 1 | Internal clock: counts on PCLK/1024 clock |

Note: • This setting is ignored when MTU2 is in phase counting mode.

**Table 20.9 TPSC[2:0] (MTU3 and MTU4)**

| Channel | Bit 2 TPSC2 | Bit 1 TPSC1 | Bit 0 TPSC0 | Description |
|---|---|---|---|---|
| MTU3, MTU4 | 0 | 0 | 0 | Internal clock: counts on PCLK/1 clock |
| | 0 | 0 | 1 | Internal clock: counts on PCLK/4 clock |
| | 0 | 1 | 0 | Internal clock: counts on PCLK/16 clock |
| | 0 | 1 | 1 | Internal clock: counts on PCLK/64 clock |
| | 1 | 0 | 0 | Internal clock: counts on PCLK/256 clock |
| | 1 | 0 | 1 | Internal clock: counts on PCLK/1024 clock |
| | 1 | 1 | 0 | External clock: counts on MTCLKA pin input |
| | 1 | 1 | 1 | External clock: counts on MTCLKB pin input |

**Table 20.10 TPSC[1:0] (MTU5)**

| Channel | Bit 1 TPSC1 | Bit 0 TPSC0 | Description |
|---|---|---|---|
| MTU5 | 0 | 0 | Internal clock: counts on PCLK/1 clock |
| | 0 | 1 | Internal clock: counts on PCLK/4 clock |
| | 1 | 0 | Internal clock: counts on PCLK/16 clock |
| | 1 | 1 | Internal clock: counts on PCLK/64 clock |

Note: • Bits 7 to 2 are reserved in MTU5. These bits are read as 0. The write value should be 0.

## 20.2.2 Timer Mode Register (TMDR)

Address(es): MTU0.TMDR 0008 8701h, MTU1.TMDR 0008 8781h, MTU2.TMDR 0008 8801h, MTU3.TMDR 0008 8602h, MTU4.TMDR 0008 8603h

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | BFE | BFB | BFA | MD[3:0] | | | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b3 to b0 | MD[3:0] | Mode Select | These bits specify the timer operating mode. See Table 20.11 for details. | R/W |
| b4 | BFA | Buffer Operation A | 0: TGRA and TGRC operate normally<br>1: TGRA and TGRC used together for buffer operation | R/W |
| b5 | BFB | Buffer Operation B | 0: TGRB and TGRD operate normally<br>1: TGRB and TGRD used together for buffer operation | R/W |
| b6 | BFE | Buffer Operation E | 0: MTU0.TGRE and MTU0.TGRF operate normally<br>1: MTU0.TGRE and MTU0.TGRF used together for buffer operation | R/W |
| b7 | — | Reserved | This bit is read as 0. The write value should be 0. | R/W |

The TMDR register specifies the operating mode of each channel. TMDR values should be specified only while TCNT operation is stopped.

**Table 20.11 Operating Mode Setting by MD[3:0] Bits**

| Bit 3 | Bit 2 | Bit 1 | Bit 0 | |
|---|---|---|---|---|
| MD3 | MD2 | MD1 | MD0 | Description |
| 0 | 0 | 0 | 0 | Normal mode |
| 0 | 0 | 0 | 1 | Setting prohibited |
| 0 | 0 | 1 | 0 | PWM mode 1 |
| 0 | 0 | 1 | 1 | PWM mode 2*1 |
| 0 | 1 | 0 | 0 | Phase counting mode 1*2 |
| 0 | 1 | 0 | 1 | Phase counting mode 2*2 |
| 0 | 1 | 1 | 0 | Phase counting mode 3*2 |
| 0 | 1 | 1 | 1 | Phase counting mode 4*2 |
| 1 | 0 | 0 | 0 | Reset-synchronized PWM mode*3 |
| 1 | 0 | 0 | 1 | Setting prohibited |
| 1 | 0 | 1 | x | Setting prohibited |
| 1 | 1 | 0 | 0 | Setting prohibited |
| 1 | 1 | 0 | 1 | Complementary PWM mode 1 (transfer at crest)*3 |
| 1 | 1 | 1 | 0 | Complementary PWM mode 2 (transfer at trough)*3 |
| 1 | 1 | 1 | 1 | Complementary PWM mode 3 (transfer at crest and trough)*3 |

x: Don't care

Note 1. PWM mode 2 cannot be set for MTU3 and MTU4.

Note 2. Phase counting mode cannot be set for MTU0, MTU3, and MTU4.

Note 3. Reset-synchronized PWM mode and complementary PWM mode can only be set for MTU3.
When MTU3 is set to reset-synchronized PWM mode or complementary PWM mode, the MTU4 settings become ineffective and conform to the MTU3 setting, respectively. 0 should be set for MTU4.
Reset-synchronized PWM mode and complementary PWM mode cannot be set for MTU0, MTU1 and MTU2.

**BFA Bit (Buffer Operation A)**

This bit specifies normal operation for TGRA or buffered operation of the combination of TGRA and TGRC. When TGRC is used as a buffer register, TGRC input capture/output compare does not take place in modes other than complementary PWM mode, but compare match with TGRC occurs in complementary PWM mode. If a compare match occurs on MTU4 in the Tb interval in complementary PWM mode, the TGIEC bit in timer interrupt enable register (MTU4.TIER) should be cleared to 0.

When MTU3 or MTU4 is set to reset-synchronized PWM mode or complementary PWM mode, the buffer operation conforms to the MTU3 setting. Set the BFA bit in MTU4.TMDR to 0.

In MTU1 and MTU2, which have no TGRC, this bit is reserved. It is read as 0. The write value should be 0. See Figure 20.40 for an illustration of the Tb interval in complementary PWM mode.

**BFB Bit (Buffer Operation B)**

This bit specifies normal operation for TGRB or buffered operation of the combination of TGRB and TGRD. When TGRD is used as a buffer register, TGRD input capture/output compare does not take place in modes other than complementary PWM mode, but compare match with TGRD occurs in complementary PWM mode. If a compare match occurs in the Tb interval in complementary PWM mode, the TGIED bit in timer interrupt enable register 3 or 4 (MTU3.TIER or MTU4.TIER) should be cleared to 0.

When MTU3 or MTU4 is set to reset-synchronized PWM mode or complementary PWM mode, the buffer operation conforms to the MTU3 setting. Set the TMDR.BFB bit in MTU4 to 0.

In MTU1 and MTU2, which have no TGRD, this bit is reserved. It is read as 0. The write value should be 0. See Figure 20.40 for an illustration of the Tb interval in complementary PWM mode.

**BFE Bit (Buffer Operation E)**

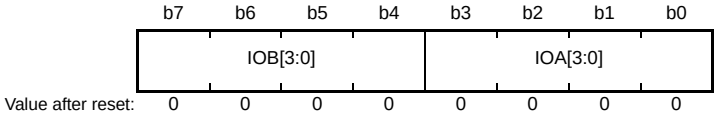
This bit specifies normal operation or buffered operation for MTU0.TGRE and MTU0.TGRF. Compare match with TGRF occurs even when TGRF is used as a buffer register.

In MTU1 to MTU4, this bit is reserved. It is read as 0. The write value should be 0.

### 20.2.3 Timer I/O Control Register (TIOR)

- MTU0.TIORH, MTU1.TIOR, MTU2.TIOR, MTU3.TIORH, MTU4.TIORH

Address(es): MTU0.TIORH 0008 8702h, MTU1.TIOR 0008 8782h, MTU2.TIOR 0008 8802h, MTU3.TIORH 0008 8604h, MTU4.TIORH 0008 8606h




| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b3 to b0 | IOA[3:0] | I/O Control A | See the following tables.*1<br>MTU0.TIORH: Table 20.20<br>MTU1.TIOR: Table 20.22<br>MTU2.TIOR: Table 20.23<br>MTU3.TIORH: Table 20.24<br>MTU4.TIORH: Table 20.26 | R/W |
| b7 to b4 | IOB[3:0] | I/O Control B | See the following tables.*1<br>MTU0.TIORH: Table 20.12<br>MTU1.TIOR: Table 20.14<br>MTU2.TIOR: Table 20.15<br>MTU3.TIORH: Table 20.16<br>MTU4.TIORH: Table 20.18 | R/W |

Note 1. If the IOn[3:0] (n = A, B) bits are changed to an “output prohibited” setting (0000b or 0100b) while output of the low or high level or toggling of the output in response to compare matches is in progress, the output becomes high impedance.

- MTU0.TIORL, MTU3.TIORL, MTU4.TIORL

Address(es): MTU0.TIORL 0008 8703h, MTU3.TIORL 0008 8605h, MTU4.TIORL 0008 8607h


| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b3 to b0 | IOC[3:0] | I/O Control C | See the following tables.*1<br>MTU0.TIORL: Table 20.21<br>MTU3.TIORL: Table 20.25<br>MTU4.TIORL: Table 20.27 | R/W |
| b7 to b4 | IOD[3:0] | I/O Control D | See the following tables.*1<br>MTU0.TIORL: Table 20.13<br>MTU3.TIORL: Table 20.17<br>MTU4.TIORL: Table 20.19 | R/W |

Note 1. If the IOn[3:0] (n = C, D) bits are changed to an “output prohibited” setting (0000b or 0100b) while output of the low or high level or toggling of the output in response to compare matches is in progress, the output becomes high impedance.

- MTU5.TIORU, MTU5.TIORV, MTU5.TIORW

Address(es): MTU5.TIORU 0008 8886h, MTU5.TIORV 0008 8896h, MTU5.TIORW 0008 88A6h

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | — | IOC[4:0] | | | | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b4 to b0 | IOC[4:0] | I/O Control C | See the following table.<br>MTU5.TIORU, MTU5.TIORV, MTU5.TIORW: Table 20.28 | R/W |
| b7 to b5 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |

The MTU has a total of 11 TIOR registers, two each for MTU0, MTU3, and MTU4, one each for MTU1 and MTU2, and three (MTU5.TIORU/V/W) for MTU5.
TIOR should be set when TMDR is set to select normal mode, PWM mode, or phase counting mode.
The initial output specified by TIOR is valid when the counter is stopped (the TSTR.CST bit is cleared to 0). Note also that, in PWM mode 2, the output at the point at which the counter is cleared to 0 is specified.
When TGRC or TGRD is designated for buffer operation, this setting is invalid and the register operates as a buffer register.

**Table 20.12 TIORH (MTU0)**

| Bit 7 | Bit 6 | Bit 5 | Bit 4 | Description | |
|---|---|---|---|---|---|
| **IOB3** | **IOB2** | **IOB1** | **IOB0** | **MTU0.TGRB Function** | **MTIOC0B Pin Function** |
| 0 | 0 | 0 | 0 | Output compare register | Output prohibited |
| 0 | 0 | 0 | 1 | | Initial output is low.<br>Low output at compare match. |
| 0 | 0 | 1 | 0 | | Initial output is low.<br>High output at compare match. |
| 0 | 0 | 1 | 1 | | Initial output is low.<br>Toggle output at compare match. |
| 0 | 1 | 0 | 0 | | Output prohibited |
| 0 | 1 | 0 | 1 | | Initial output is high.<br>Low output at compare match. |
| 0 | 1 | 1 | 0 | | Initial output is high.<br>High output at compare match. |
| 0 | 1 | 1 | 1 | | Initial output is high.<br>Toggle output at compare match. |
| 1 | 0 | 0 | 0 | Input capture register | Input capture at rising edge. |
| 1 | 0 | 0 | 1 | | Input capture at falling edge. |
| 1 | 0 | 1 | x | | Input capture at both edges. |
| 1 | 1 | x | x | | Capture input source is count clock in MTU1.<br>Input capture at MTU1.TCNT up-count/down-count. |

x: Don't care

**Table 20.13 TIORL (MTU0)**

| Bit 7 | Bit 6 | Bit 5 | Bit 4 | Description | |
|---|---|---|---|---|---|
| **IOD3** | **IOD2** | **IOD1** | **IOD0** | **MTU0.TGRD Function** | **MTIOC0D Pin Function** |
| 0 | 0 | 0 | 0 | Output compare register*1 | Output prohibited |
| 0 | 0 | 0 | 1 | | Initial output is low.<br>Low output at compare match. |
| 0 | 0 | 1 | 0 | | Initial output is low.<br>High output at compare match. |
| 0 | 0 | 1 | 1 | | Initial output is low.<br>Toggle output at compare match. |
| 0 | 1 | 0 | 0 | | Output prohibited |
| 0 | 1 | 0 | 1 | | Initial output is high.<br>Low output at compare match. |
| 0 | 1 | 1 | 0 | | Initial output is high.<br>High output at compare match. |
| 0 | 1 | 1 | 1 | | Initial output is high.<br>Toggle output at compare match. |
| 1 | 0 | 0 | 0 | Input capture register*1 | Input capture at rising edge. |
| 1 | 0 | 0 | 1 | | Input capture at falling edge. |
| 1 | 0 | 1 | x | | Input capture at both edges. |
| 1 | 1 | x | x | | Capture input source is count clock in MTU1.<br>Input capture at MTU1.TCNT up-count/down-count. |

x: Don't care

Note 1. When the MTU0.TMDR.BFB is set to 1 and MTU0.TGRD is used as a buffer register, this setting is invalid and input capture/output compare is not generated.

**Table 20.14 TIOR (MTU1)**

| Bit 7 | Bit 6 | Bit 5 | Bit 4 | Description | |
|---|---|---|---|---|---|
| **IOB3** | **IOB2** | **IOB1** | **IOB0** | **MTU1.TGRB Function** | **MTIOC1B Pin Function** |
| 0 | 0 | 0 | 0 | MTU1.TGRB works as an output compare register | Output prohibited |
| 0 | 0 | 0 | 1 | | Initial output is low.<br>Low output at compare match. |
| 0 | 0 | 1 | 0 | | Initial output is low.<br>High output at compare match. |
| 0 | 0 | 1 | 1 | | Initial output is low.<br>Toggle output at compare match. |
| 0 | 1 | 0 | 0 | | Output prohibited |
| 0 | 1 | 0 | 1 | | Initial output is high.<br>Low output at compare match. |
| 0 | 1 | 1 | 0 | | Initial output is high.<br>High output at compare match. |
| 0 | 1 | 1 | 1 | | Initial output is high.<br>Toggle output at compare match. |
| 1 | 0 | 0 | 0 | Input capture register | Input capture at rising edge. |
| 1 | 0 | 0 | 1 | | Input capture at falling edge. |
| 1 | 0 | 1 | x | | Input capture at both edges. |
| 1 | 1 | x | x | | Input capture at generation of MTU0.TGRC compare match/input capture. |

x: Don't care

**Table 20.15 TIOR (MTU2)**

| Bit 7 | Bit 6 | Bit 5 | Bit 4 | Description | |
|---|---|---|---|---|---|
| **IOB3** | **IOB2** | **IOB1** | **IOB0** | **MTU2.TGRB Function** | **MTIOC2B Pin Function** |
| 0 | 0 | 0 | 0 | MTU2.TGRB works as an output compare register | Output prohibited |
| 0 | 0 | 0 | 1 | | Initial output is low.<br>Low output at compare match. |
| 0 | 0 | 1 | 0 | | Initial output is low.<br>High output at compare match. |
| 0 | 0 | 1 | 1 | | Initial output is low.<br>Toggle output at compare match. |
| 0 | 1 | 0 | 0 | | Output prohibited |
| 0 | 1 | 0 | 1 | | Initial output is high.<br>Low output at compare match. |
| 0 | 1 | 1 | 0 | | Initial output is high.<br>High output at compare match. |
| 0 | 1 | 1 | 1 | | Initial output is high.<br>Toggle output at compare match. |
| 1 | x | 0 | 0 | Input capture register | Input capture at rising edge. |
| 1 | x | 0 | 1 | | Input capture at falling edge. |
| 1 | x | 1 | x | | Input capture at both edges. |

x: Don't care

**Table 20.16 TIORH (MTU3)**

| Bit 7 | Bit 6 | Bit 5 | Bit 4 | Description | |
|---|---|---|---|---|---|
| **IOB3** | **IOB2** | **IOB1** | **IOB0** | **MTU3.TGRB Function** | **MTIOC3B Pin Function** |
| 0 | 0 | 0 | 0 | Output compare register | Output prohibited |
| 0 | 0 | 0 | 1 | | Initial output is low.<br>Low output at compare match. |
| 0 | 0 | 1 | 0 | | Initial output is low.<br>High output at compare match. |
| 0 | 0 | 1 | 1 | | Initial output is low.<br>Toggle output at compare match. |
| 0 | 1 | 0 | 0 | | Output prohibited |
| 0 | 1 | 0 | 1 | | Initial output is high.<br>Low output at compare match. |
| 0 | 1 | 1 | 0 | | Initial output is high.<br>High output at compare match. |
| 0 | 1 | 1 | 1 | | Initial output is high.<br>Toggle output at compare match. |
| 1 | x | 0 | 0 | Input capture register | Input capture at rising edge. |
| 1 | x | 0 | 1 | | Input capture at falling edge. |
| 1 | x | 1 | x | | Input capture at both edges. |

x: Don't care

**Table 20.17 TIORL (MTU3)**

| Bit 7 | Bit 6 | Bit 5 | Bit 4 | Description | |
|---|---|---|---|---|---|
| **IOD3** | **IOD2** | **IOD1** | **IOD0** | **MTU3.TGRD Function** | **MTIOC3D Pin Function** |
| 0 | 0 | 0 | 0 | Output compare register*1 | Output prohibited |
| 0 | 0 | 0 | 1 | | Initial output is low.<br>Low output at compare match. |
| 0 | 0 | 1 | 0 | | Initial output is low.<br>High output at compare match. |
| 0 | 0 | 1 | 1 | | Initial output is low.<br>Toggle output at compare match. |
| 0 | 1 | 0 | 0 | | Output prohibited |
| 0 | 1 | 0 | 1 | | Initial output is high.<br>Low output at compare match. |
| 0 | 1 | 1 | 0 | | Initial output is high.<br>High output at compare match. |
| 0 | 1 | 1 | 1 | | Initial output is high.<br>Toggle output at compare match. |
| 1 | x | 0 | 0 | Input capture register*1 | Input capture at rising edge. |
| 1 | x | 0 | 1 | | Input capture at falling edge. |
| 1 | x | 1 | x | | Input capture at both edges. |

x: Don't care

Note 1. When the BFB bit in MTU3.TMDR is set to 1 and MTU3.TGRD is used as a buffer register, this setting is invalid and input capture/output compare is not generated.

**Table 20.18 TIORH (MTU4)**

| Bit 7 | Bit 6 | Bit 5 | Bit 4 | Description | |
|---|---|---|---|---|---|
| **IOB3** | **IOB2** | **IOB1** | **IOB0** | **MTU4.TGRB Function** | **MTIOC4B Pin Function** |
| 0 | 0 | 0 | 0 | MTU4.TGRB works as an output compare register | Output prohibited |
| 0 | 0 | 0 | 1 | | Initial output is low.<br>Low output at compare match. |
| 0 | 0 | 1 | 0 | | Initial output is low.<br>High output at compare match. |
| 0 | 0 | 1 | 1 | | Initial output is low.<br>Toggle output at compare match. |
| 0 | 1 | 0 | 0 | | Output prohibited |
| 0 | 1 | 0 | 1 | | Initial output is high.<br>Low output at compare match. |
| 0 | 1 | 1 | 0 | | Initial output is high.<br>High output at compare match. |
| 0 | 1 | 1 | 1 | | Initial output is high.<br>Toggle output at compare match. |
| 1 | x | 0 | 0 | Input capture register | Input capture at rising edge. |
| 1 | x | 0 | 1 | | Input capture at falling edge. |
| 1 | x | 1 | x | | Input capture at both edges. |

x: Don't care

**Table 20.19 TIORL (MTU4)**

| Bit 7 | Bit 6 | Bit 5 | Bit 4 | Description | |
|---|---|---|---|---|---|
| **IOD3** | **IOD2** | **IOD1** | **IOD0** | **MTU4.TGRD Function** | **MTIOC4D Pin Function** |
| 0 | 0 | 0 | 0 | Output compare register*1 | Output prohibited |
| 0 | 0 | 0 | 1 | | Initial output is low.<br>Low output at compare match. |
| 0 | 0 | 1 | 0 | | Initial output is low.<br>High output at compare match. |
| 0 | 0 | 1 | 1 | | Initial output is low.<br>Toggle output at compare match. |
| 0 | 1 | 0 | 0 | | Output prohibited |
| 0 | 1 | 0 | 1 | | Initial output is high.<br>Low output at compare match. |
| 0 | 1 | 1 | 0 | | Initial output is high.<br>High output at compare match. |
| 0 | 1 | 1 | 1 | | Initial output is high.<br>Toggle output at compare match. |
| 1 | x | 0 | 0 | Input capture register*1 | Input capture at rising edge. |
| 1 | x | 0 | 1 | | Input capture at falling edge. |
| 1 | x | 1 | x | | Input capture at both edges. |

x: Don't care

Note 1. When the BFB bit in MTU4.TMDR is set to 1 and MTU4.TGRD is used as a buffer register, this setting is invalid and input capture/output compare is not generated.

**Table 20.20 TIORH (MTU0)**

| Bit 3 | Bit 2 | Bit 1 | Bit 0 | Description | |
|---|---|---|---|---|---|
| **IOA3** | **IOA2** | **IOA1** | **IOA0** | **MTU0.TGRA Function** | **MTIOC0A Pin Function** |
| 0 | 0 | 0 | 0 | Output compare register | Output prohibited |
| 0 | 0 | 0 | 1 | | Initial output is low.<br>Low output at compare match. |
| 0 | 0 | 1 | 0 | | Initial output is low.<br>High output at compare match. |
| 0 | 0 | 1 | 1 | | Initial output is low.<br>Toggle output at compare match. |
| 0 | 1 | 0 | 0 | | Output prohibited |
| 0 | 1 | 0 | 1 | | Initial output is high.<br>Low output at compare match. |
| 0 | 1 | 1 | 0 | | Initial output is high.<br>High output at compare match. |
| 0 | 1 | 1 | 1 | | Initial output is high.<br>Toggle output at compare match. |
| 1 | 0 | 0 | 0 | Input capture register | Input capture at rising edge. |
| 1 | 0 | 0 | 1 | | Input capture at falling edge. |
| 1 | 0 | 1 | x | | Input capture at both edges. |
| 1 | 1 | x | x | | Capture input source is count clock in MTU1.<br>Input capture at MTU1.TCNT up-count/down-count. |

x: Don't care

**Table 20.21 TIORL (MTU0)**

| Bit 3 | Bit 2 | Bit 1 | Bit 0 | Description | |
|---|---|---|---|---|---|
| **IOC3** | **IOC2** | **IOC1** | **IOC0** | **MTU0.TGRC Function** | **MTIOC0C Pin Function** |
| 0 | 0 | 0 | 0 | Output compare register*1 | Output prohibited |
| 0 | 0 | 0 | 1 | | Initial output is low.<br>Low output at compare match. |
| 0 | 0 | 1 | 0 | | Initial output is low.<br>High output at compare match. |
| 0 | 0 | 1 | 1 | | Initial output is low.<br>Toggle output at compare match. |
| 0 | 1 | 0 | 0 | | Output prohibited |
| 0 | 1 | 0 | 1 | | Initial output is high.<br>Low output at compare match. |
| 0 | 1 | 1 | 0 | | Initial output is high.<br>High output at compare match. |
| 0 | 1 | 1 | 1 | | Initial output is high.<br>Toggle output at compare match. |
| 1 | 0 | 0 | 0 | Input capture register*1 | Input capture at rising edge. |
| 1 | 0 | 0 | 1 | | Input capture at falling edge. |
| 1 | 0 | 1 | x | | Input capture at both edges. |
| 1 | 1 | x | x | | Capture input source is count clock in MTU1.<br>Input capture at MTU1.TCNT up-count/down-count. |

x: Don't care

Note 1. When the BFA bit in MTU0.TMDR is set to 1 and MTU0.TGRC is used as a buffer register, this setting is invalid and input capture/output compare is not generated.

**Table 20.22 TIOR (MTU1)**

| Bit 3 | Bit 2 | Bit 1 | Bit 0 | Description | |
|---|---|---|---|---|---|
| **IOA3** | **IOA2** | **IOA1** | **IOA0** | **MTU1.TGRA Function** | **MTIOC1A Pin Function** |
| 0 | 0 | 0 | 0 | Output compare register | Output prohibited |
| 0 | 0 | 0 | 1 | | Initial output is low.<br>Low output at compare match. |
| 0 | 0 | 1 | 0 | | Initial output is low.<br>High output at compare match. |
| 0 | 0 | 1 | 1 | | Initial output is low.<br>Toggle output at compare match. |
| 0 | 1 | 0 | 0 | | Output prohibited |
| 0 | 1 | 0 | 1 | | Initial output is high.<br>Low output at compare match. |
| 0 | 1 | 1 | 0 | | Initial output is high.<br>High output at compare match. |
| 0 | 1 | 1 | 1 | | Initial output is high.<br>Toggle output at compare match. |
| 1 | 0 | 0 | 0 | Input capture register | Input capture at rising edge. |
| 1 | 0 | 0 | 1 | | Input capture at falling edge. |
| 1 | 0 | 1 | x | | Input capture at both edges. |
| 1 | 1 | x | x | | Input capture at generation of MTU0.TGRA compare match/input capture. |

x: Don't care

**Table 20.23 TIOR (MTU2)**

| Bit 3 | Bit 2 | Bit 1 | Bit 0 | Description | |
|---|---|---|---|---|---|
| **IOA3** | **IOA2** | **IOA1** | **IOA0** | **MTU2.TGRA Function** | **MTIOC2A Pin Function** |
| 0 | 0 | 0 | 0 | Output compare register | Output prohibited |
| 0 | 0 | 0 | 1 | | Initial output is low.<br>Low output at compare match. |
| 0 | 0 | 1 | 0 | | Initial output is low.<br>High output at compare match. |
| 0 | 0 | 1 | 1 | | Initial output is low.<br>Toggle output at compare match. |
| 0 | 1 | 0 | 0 | | Output prohibited |
| 0 | 1 | 0 | 1 | | Initial output is high.<br>Low output at compare match. |
| 0 | 1 | 1 | 0 | | Initial output is high.<br>High output at compare match. |
| 0 | 1 | 1 | 1 | | Initial output is high.<br>Toggle output at compare match. |
| 1 | x | 0 | 0 | Input capture register | Input capture at rising edge. |
| 1 | x | 0 | 1 | | Input capture at falling edge. |
| 1 | x | 1 | x | | Input capture at both edges. |

x: Don't care

**Table 20.24 TIORH (MTU3)**

| Bit 3 | Bit 2 | Bit 1 | Bit 0 | Description | |
|---|---|---|---|---|---|
| **IOA3** | **IOA2** | **IOA1** | **IOA0** | **MTU3.TGRA Function** | **MTIOC3A Pin Function** |
| 0 | 0 | 0 | 0 | Output compare register | Output prohibited |
| 0 | 0 | 0 | 1 | | Initial output is low.<br>Low output at compare match. |
| 0 | 0 | 1 | 0 | | Initial output is low.<br>High output at compare match. |
| 0 | 0 | 1 | 1 | | Initial output is low.<br>Toggle output at compare match. |
| 0 | 1 | 0 | 0 | | Output prohibited |
| 0 | 1 | 0 | 1 | | Initial output is high.<br>Low output at compare match. |
| 0 | 1 | 1 | 0 | | Initial output is high.<br>High output at compare match. |
| 0 | 1 | 1 | 1 | | Initial output is high.<br>Toggle output at compare match. |
| 1 | x | 0 | 0 | Input capture register | Input capture at rising edge. |
| 1 | x | 0 | 1 | | Input capture at falling edge. |
| 1 | x | 1 | x | | Input capture at both edges. |

x: Don't care

**Table 20.25 TIORL (MTU3)**

| Bit 3 | Bit 2 | Bit 1 | Bit 0 | Description | |
|---|---|---|---|---|---|
| **IOC3** | **IOC2** | **IOC1** | **IOC0** | **MTU3.TGRC Function** | **MTIOC3C Pin Function** |
| 0 | 0 | 0 | 0 | Output compare register*1 | Output prohibited |
| 0 | 0 | 0 | 1 | | Initial output is low.<br>Low output at compare match. |
| 0 | 0 | 1 | 0 | | Initial output is low.<br>High output at compare match. |
| 0 | 0 | 1 | 1 | | Initial output is low.<br>Toggle output at compare match. |
| 0 | 1 | 0 | 0 | | Output prohibited |
| 0 | 1 | 0 | 1 | | Initial output is high.<br>Low output at compare match. |
| 0 | 1 | 1 | 0 | | Initial output is high.<br>High output at compare match. |
| 0 | 1 | 1 | 1 | | Initial output is high.<br>Toggle output at compare match. |
| 1 | x | 0 | 0 | Input capture register*1 | Input capture at rising edge. |
| 1 | x | 0 | 1 | | Input capture at falling edge. |
| 1 | x | 1 | x | | Input capture at both edges. |

x: Don't care

Note 1. When the BFA bit in MTU3.TMDR is set to 1 and MTU3.TGRC is used as a buffer register, this setting is invalid and input capture/output compare is not generated.

**Table 20.26 TIORH (MTU4)**

| Bit 3 | Bit 2 | Bit 1 | Bit 0 | Description | |
|---|---|---|---|---|---|
| **IOA3** | **IOA2** | **IOA1** | **IOA0** | **MTU4.TGRA Function** | **MTIOC4A Pin Function** |
| 0 | 0 | 0 | 0 | Output compare register | Output prohibited |
| 0 | 0 | 0 | 1 | | Initial output is low.<br>Low output at compare match. |
| 0 | 0 | 1 | 0 | | Initial output is low.<br>High output at compare match. |
| 0 | 0 | 1 | 1 | | Initial output is low.<br>Toggle output at compare match. |
| 0 | 1 | 0 | 0 | | Output prohibited |
| 0 | 1 | 0 | 1 | | Initial output is high.<br>Low output at compare match. |
| 0 | 1 | 1 | 0 | | Initial output is high.<br>High output at compare match. |
| 0 | 1 | 1 | 1 | | Initial output is high.<br>Toggle output at compare match. |
| 1 | x | 0 | 0 | Input capture register | Input capture at rising edge. |
| 1 | x | 0 | 1 | | Input capture at falling edge. |
| 1 | x | 1 | x | | Input capture at both edges. |

x: Don't care

**Table 20.27 TIORL (MTU4)**

| Bit 3 | Bit 2 | Bit 1 | Bit 0 | Description | |
|---|---|---|---|---|---|
| **IOC3** | **IOC2** | **IOC1** | **IOC0** | **MTU4.TGRC Function** | **MTIOC4C Pin Function** |
| 0 | 0 | 0 | 0 | Output compare register*1 | Output prohibited |
| 0 | 0 | 0 | 1 | | Initial output is low.<br>Low output at compare match. |
| 0 | 0 | 1 | 0 | | Initial output is low.<br>High output at compare match. |
| 0 | 0 | 1 | 1 | | Initial output is low.<br>Toggle output at compare match. |
| 0 | 1 | 0 | 0 | | Output prohibited |
| 0 | 1 | 0 | 1 | | Initial output is high.<br>Low output at compare match. |
| 0 | 1 | 1 | 0 | | Initial output is high.<br>High output at compare match. |
| 0 | 1 | 1 | 1 | | Initial output is high.<br>Toggle output at compare match. |
| 1 | x | 0 | 0 | Input capture register*1 | Input capture at rising edge. |
| 1 | x | 0 | 1 | | Input capture at falling edge. |
| 1 | x | 1 | x | | Input capture at both edges. |

x: Don't care

Note 1. When the BFA bit in MTU4.TMDR is set to 1 and MTU4.TGRC is used as a buffer register, this setting is invalid and input capture/output compare is not generated.

**Table 20.28 TIORU, TIORV, and TIORW (MTU5)**

| Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Description | |
|---|---|---|---|---|---|---|
| IOC4 | IOC3 | IOC2 | IOC1 | IOC0 | MTU5.TGRU, MTU5.TGRV, MTU5.TGRW Function | MTIC5U, MTIC5V, MTIC5W Pin Function |
| 0 | 0 | 0 | 0 | 0 | Compare match register | Compare match |
| 0 | 0 | 0 | 0 | 1 | | Setting prohibited |
| 0 | 0 | 0 | 1 | x | | Setting prohibited |
| 0 | 0 | 1 | x | x | | Setting prohibited |
| 0 | 1 | x | x | x | | Setting prohibited |
| 1 | 0 | 0 | 0 | 0 | Input capture register | Setting prohibited |
| 1 | 0 | 0 | 0 | 1 | | Input capture at rising edge. |
| 1 | 0 | 0 | 1 | 0 | | Input capture at falling edge. |
| 1 | 0 | 0 | 1 | 1 | | Input capture at both edges. |
| 1 | 0 | 1 | x | x | | Setting prohibited |
| 1 | 1 | 0 | 0 | 0 | | Setting prohibited |
| 1 | 1 | 0 | 0 | 1 | | Measurement of low pulse width of external input signal. Capture at trough in complementary PWM mode. |
| 1 | 1 | 0 | 1 | 0 | | Measurement of low pulse width of external input signal. Capture at trough in complementary PWM mode. |
| 1 | 1 | 0 | 1 | 1 | | Measurement of low pulse width of external input signal. Capture at trough in complementary PWM mode. |
| 1 | 1 | 1 | 0 | 0 | | Setting prohibited |
| 1 | 1 | 1 | 0 | 1 | | Measurement of high pulse width of external input signal. Capture at trough in complementary PWM mode. |
| 1 | 1 | 1 | 1 | 0 | | Measurement of high pulse width of external input signal. Capture at trough in complementary PWM mode. |
| 1 | 1 | 1 | 1 | 1 | | Measurement of high pulse width of external input signal. Capture at trough in complementary PWM mode. |

x: Don't care

## 20.2.4 Timer Compare Match Clear Register (TCNTCMPCLR)

Address(es): MTU5.TCNTCMPCLR 0008 88B6h

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | CMPCLR5U | CMPCLR5V | CMPCLR5W |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b0 | CMPCLR5W | TCNT Compare Clear 5W | 0: Disables MTU5.TCNTW to be cleared to 0000h at MTU5.TCNTW and MTU5.TGRW compare match or input capture<br>1: Enables MTU5.TCNTW to be cleared to 0000h at MTU5.TCNTW and MTU5.TGRW compare match or input capture | R/W |
| b1 | CMPCLR5V | TCNT Compare Clear 5V | 0: Disables MTU5.TCNTV to be cleared to 0000h at MTU5.TCNTV and MTU5.TGRV compare match or input capture<br>1: Enables MTU5.TCNTV to be cleared to 0000h at MTU5.TCNTV and MTU5.TGRV compare match or input capture | R/W |
| b2 | CMPCLR5U | TCNT Compare Clear 5U | 0: Disables MTU5.TCNTU to be cleared to 0000h at MTU5.TCNTU and MTU5.TGRU compare match or input capture<br>1: Enables MTU5.TCNTU to be cleared to 0000h at MTU5.TCNTU and MTU5.TGRU compare match or input capture | R/W |
| b7 to b3 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |

The TCNTCMPCLR register specifies requests to clear MTU5.TCNTU, MTU5.TCNTV, and MTU5.TCNTW.

## 20.2.5 Timer Interrupt Enable Register (TIER)

- TIER (MTU0 to MTU4)

Address(es): MTU0.TIER 0008 8704h, MTU1.TIER 0008 8784h, MTU2.TIER 0008 8804h, MTU3.TIER 0008 8608h, MTU4.TIER 0008 8609h

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | TTGE | TTGE2 | TCIEU | TCIEV | TGIED | TGIEC | TGIEB | TGIEA |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b0 | TGIEA | TGR Interrupt Enable A | 0: Interrupt requests (TGIA) disabled<br>1: Interrupt requests (TGIA) enabled | R/W |
| b1 | TGIEB | TGR Interrupt Enable B | 0: Interrupt requests (TGIB) disabled<br>1: Interrupt requests (TGIB) enabled | R/W |
| b2 | TGIEC | TGR Interrupt Enable C | 0: Interrupt requests (TGIC) disabled<br>1: Interrupt requests (TGIC) enabled | R/W |
| b3 | TGIED | TGR Interrupt Enable D | 0: Interrupt requests (TGID) disabled<br>1: Interrupt requests (TGID) enabled | R/W |
| b4 | TCIEV | Overflow Interrupt Enable | 0: Interrupt requests (TCIV) disabled<br>1: Interrupt requests (TCIV) enabled | R/W |
| b5 | TCIEU | Underflow Interrupt Enable | 0: Interrupt requests (TCIU) disabled<br>1: Interrupt requests (TCIU) enabled | R/W |
| b6 | TTGE2 | A/D Converter Start Request Enable 2 | 0: A/D converter start request generation by MTU4.TCNT underflow (trough) disabled<br>1: A/D converter start request generation by MTU4.TCNT underflow (trough) enabled | R/W |
| b7 | TTGE | A/D Converter Start Request Enable | 0: A/D converter start request generation disabled<br>1: A/D converter start request generation enabled | R/W |

The MTU has a total of seven TIER registers, two each for MTU0 and one each for MTU1 to MTU5.
The TIER register enables or disables interrupt requests in each channel.

### TGIEA and TGIEB Bits (TGR Interrupt Enable A and B)

Each bit enables or disables interrupt requests (TGIn) (n = A, B).

### TGIEC and TGIED Bits (TGR Interrupt Enable C and D)

Each bit enables or disables interrupt requests (TGIn) in MTU0, MTU3 and MTU4 (n = C, D).
In MTU1 and MTU2, these bits are reserved. They are read as 0. The write value should be 0.

### TCIEV Bit (Overflow Interrupt Enable)

This bit enables or disables interrupt requests (TCIV).

### TCIEU Bit (Underflow Interrupt Enable)

This bit enables or disables interrupt requests (TCIU) in MTU1 and MTU2.
In MTU0, MTU3, and MTU4, this bit is reserved. It is read as 0. The write value should be 0.

### TTGE2 Bit (A/D Converter Start Request Enable 2)

This bit enables or disables generation of A/D converter start requests by MTU4.TCNT underflow (trough) in complementary PWM mode.
In MTU0 to MTU3, this bit is reserved. It is read as 0. The write value should be 0.

### TTGE Bit (A/D Converter Start Request Enable)

This bit enables or disables generation of A/D converter start requests by TGRA input capture/compare match.

- TIER2 (MTU0)

Address(es): MTU0.TIER2 0008 8724h

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | TGIEF | TGIEE |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b0 | TGIEE | TGR Interrupt Enable E | 0: Interrupt requests (TGIE) disabled<br>1: Interrupt requests (TGIE) enabled | R/W |
| b1 | TGIEF | TGR Interrupt Enable F | 0: Interrupt requests (TGIF) disabled<br>1: Interrupt requests (TGIF) enabled | R/W |
| b7 to b2 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |

### TGIEE and TGIEF Bits (TGR Interrupt Enable E and F)

Each bit enables or disables interrupt requests by compare match between MTU0.TCNT and MTU0.TGRn (n = E, F).

- TIER (MTU5)

Address(es): MTU5.TIER 0008 88B2h

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | TGIE5U | TGIE5V | TGIE5W |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b0 | TGIE5W | TGR Interrupt Enable 5W | 0: Interrupt requests TGI5W disabled<br>1: Interrupt requests TGI5W enabled | R/W |
| b1 | TGIE5V | TGR Interrupt Enable 5V | 0: Interrupt requests TGI5V disabled<br>1: Interrupt requests TGI5V enabled | R/W |
| b2 | TGIE5U | TGR Interrupt Enable 5U | 0: Interrupt requests TGI5U disabled<br>1: Interrupt requests TGI5U enabled | R/W |
| b7 to b3 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |

### TGIE5W, TGIE5V, and TGIE5U Bits (TGR Interrupt Enable 5n)

Each bit enables or disables interrupt requests (TGI5n) (n = W, V, or U).

## 20.2.6 Timer Status Register (TSR)

- TSR (MTU0 to MTU4)

Address(es): MTU0.TSR 0008 8705h, MTU1.TSR 0008 8785h, MTU2.TSR 0008 8805h, MTU3.TSR 0008 862Ch, MTU4.TSR 0008 862Dh

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | TCFD | — | — | — | — | — | — | — |
| Value after reset: | 1 | 1 | x | x | x | x | x | x |

x: Undefined

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b5 to b0 | — | Reserved | These bits are read as undefined. The write value should be 1. | R/W |
| b6 | — | Reserved | This bit is read as 1. The write value should be 1. | R/W |
| b7 | TCFD | Count Direction Flag | 0: TCNT counts down<br>1: TCNT counts up | R |

The MTU has a total of five TSR registers, one each for MTU0 to MTU4.
The TSR register indicates the status of each channel.

**TCFD Flag (Count Direction Flag)**

Status flag that shows the direction in which TCNT counts in MTU1 to MTU4.
In MTU0, this bit is reserved. It is read as 1. The write value should be 1.

## 20.2.7 Timer Buffer Operation Transfer Mode Register (TBTM)

Address(es): MTU0.TBTM 0008 8726h, MTU3.TBTM 0008 8638h, MTU4.TBTM 0008 8639h

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | TTSE | TTSB | TTSA |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b0 | TTSA | Timing Select A | 0: When compare match A occurs in each channel, data is transferred from TGRC to TGRA<br>1: When TCNT is cleared in each channel, data is transferred from TGRC to TGRA | R/W |
| b1 | TTSB | Timing Select B | 0: When compare match B occurs in each channel, data is transferred from TGRD to TGRB<br>1: When TCNT is cleared in each channel, data is transferred from TGRD to TGRB | R/W |
| b2 | TTSE | Timing Select E | 0: When compare match E occurs in MTU0, data is transferred from MTU0.TGRF to MTU0.TGRE<br>1: When MTU0.TCNT is cleared in MTU0, data is transferred from MTU0.TGRF to MTU0.TGRE | R/W |
| b7 to b3 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |

The MTU has a total of three TBTM registers, one each for MTU0, MTU3 and MTU4.
The TBTM register specifies the timing for transferring data from the buffer register to the timer general register in PWM mode.

### TTSA Bit (Timing Select A)

This bit specifies the timing for transferring data from TGRC to TGRA in each channel when they are used together for buffer operation. When a channel is not set to PWM mode, do not set the TTSA bit in the channel to 1.

### TTSB Bit (Timing Select B)

This bit specifies the timing for transferring data from TGRD to TGRB in each channel when they are used together for buffer operation. When a channel is not set to PWM mode, do not set the TTSB bit in the channel to 1.

### TTSE Bit (Timing Select E)

This bit specifies the timing for transferring data from MTU0.TGRF to MTU0.TGRE when they are used together for buffer operation. In MTU3 and MTU4, this bit is read as 0. The write value should be 0. When MTU0 is not set to PWM mode, do not set the TTSE bit to 1.

## 20.2.8 Timer Input Capture Control Register (TICCR)

Address(es): MTU1.TICCR 0008 8790h

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | I2BE | I2AE | I1BE | I1AE |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b0 | I1AE | Input Capture Enable | 0: Does not include the MTIOC1A pin in the MTU2.TGRA input capture conditions<br>1: Includes the MTIOC1A pin in the MTU2.TGRA input capture conditions | R/W |
| b1 | I1BE | Input Capture Enable | 0: Does not include the TMTIOC1B pin in the MTU2.TGRB input capture conditions<br>1: Includes the MTIOC1B pin in the MTU2.TGRB input capture conditions | R/W |
| b2 | I2AE | Input Capture Enable | 0: Does not include the MTIOC2A pin in the MTU1.TGRA input capture conditions<br>1: Includes the MTIOC2A pin in the MTU1.TGRA input capture conditions | R/W |
| b3 | I2BE | Input Capture Enable | 0: Does not include the MTIOC2B pin in the MTU1.TGRB input capture conditions<br>1: Includes the MTIOC2B pin in the MTU1.TGRB input capture conditions | R/W |
| b7 to b4 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |

The MTU has one TICCR for MTU1.

The TICCR register specifies input capture conditions when MTU1.TCNT and MTU2.TCNT are cascaded.

## 20.2.9 Timer A/D Converter Start Request Control Register (TADCR)

Address(es): MTU4.TADCR 0008 8640h

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | BF[1:0] | | — | — | — | — | — | — | UT4AE | DT4AE | UT4BE | DT4BE | ITA3AE | ITA4VE | ITB3AE | ITB4VE |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b0 | ITB4VE | TCIV4 Interrupt Skipping Link Enable | 0: TCI4V interrupt skipping is not linked<br>1: TCI4V interrupt skipping is linked | R/W*1 |
| b1 | ITB3AE | TGIA3 Interrupt Skipping Link Enable | 0: TGI3A interrupt skipping is not linked<br>1: TGI3A interrupt skipping is linked | R/W*1 |
| b2 | ITA4VE | TCIV4 Interrupt Skipping Link Enable | 0: TCI4V interrupt skipping is not linked<br>1: TCI4V interrupt skipping is linked | R/W*1 |
| b3 | ITA3AE | TGIA3 Interrupt Skipping Link Enable | 0: TGI3A interrupt skipping is not linked<br>1: TGI3A interrupt skipping is linked | R/W*1 |
| b4 | DT4BE | Down-Count TRG4BN Enable | 0: A/D converter start requests (TRG4BN) disabled during MTU4.TCNT down-count operation<br>1: A/D converter start requests (TRG4BN) enabled during MTU4.TCNT down-count operation | R/W*1 |
| b5 | UT4BE | Up-Count TRG4BN Enable | 0: A/D converter start requests (TRG4BN) disabled during MTU4.TCNT up-count operation<br>1: A/D converter start requests (TRG4BN) enabled during MTU4.TCNT up-count operation | R/W |
| b6 | DT4AE | Down-Count TRG4AN Enable | 0: A/D converter start requests (TRG4AN) disabled during MTU4.TCNT down-count operation<br>1: A/D converter start requests (TRG4AN) enabled during MTU4.TCNT down-count operation | R/W*1 |
| b7 | UT4AE | Up-Count TRG4AN Enable | 0: A/D converter start requests (TRG4AN) disabled during MTU4.TCNT up-count operation<br>1: A/D converter start requests (TRG4AN) enabled during MTU4.TCNT up-count operation | R/W |
| b13 to b8 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |
| b15, b14 | BF[1:0] | MTU4.TADCOBRA/TADCOBRB Transfer Timing Select | See Table 20.29 for details. | R/W |

Note: • TADCR must not be accessed in 8-bit units; it should be accessed in 16-bit units.
Note: • When interrupt skipping is disabled (the T3AEN and T4VEN bits in the timer interrupt skipping set register (TITCR) are cleared to 0 or the interrupt skipping count setting bits (T3ACOR and T4VCOR) in TITCR are cleared to 0), do not link A/D converter start requests with interrupt skipping operation (clear the ITA3AE, ITA4VE, ITB3AE, and ITB4VE bits in the timer A/D converter start request control register (TADCR) to 0).
Note: • If link with interrupt skipping is enabled while interrupt skipping is disabled, A/D converter start requests will not be issued.
Note 1. Do not set any bit from among b6 and b4 to b0 to 1 unless complementary PWM mode is not selected.

The TADCR register enables or disables A/D converter start requests and specifies whether to link A/D converter start requests with interrupt skipping operation.

**Table 20.29 Setting of Transfer Timing by BF[1:0] Bits**

| Bit 15 | Bit 14 | |
|---|---|---|
| BF1 | BF0 | Description |
| 0 | 0 | Does not transfer data from the cycle set buffer register to the cycle set register. |
| 0 | 1 | Transfers data from the cycle set buffer register to the cycle set register at the crest of the MTU4.TCNT count.*1 |
| 1 | 0 | Transfers data from the cycle set buffer register to the cycle set register at the trough of the MTU4.TCNT count.*2 |
| 1 | 1 | Transfers data from the cycle set buffer register to the cycle set register at the crest and trough of the MTU4.TCNT count.*2 |

Note 1. Data is transferred from the cycle set buffer register to the cycle set register when the crest of the MTU4.TCNT count is reached in complementary PWM mode, when a compare match occurs between MTU3.TCNT and MTU3.TGRA in reset-synchronized PWM mode, or when a compare match occurs between MTU4.TCNT and MTU4.TGRA in PWM mode 1 or normal operation mode.

Note 2. These settings are prohibited when complementary PWM mode is not selected.

## 20.2.10 Timer A/D Converter Start Request Cycle Set Registers A and B (TADCORA and TADCORB)

Address(es): MTU4.TADCORA 0008 8644h, MTU4.TADCORB 0008 8646h

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Value after reset: | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

Note: • MTU4.TADCORA and MTU4.TADCORB must not be accessed in 8-bit units; they should be accessed in 16-bit units.

The TADCOBRA and TADCOBRB registers specify the A/D converter start request cycle. When the MTU4.TCNT count reaches the value in TADCORA or TADCORB, a corresponding A/D converter start request will be issued.

## 20.2.11 Timer A/D Converter Start Request Cycle Set Buffer Registers A and B (TADCOBRA and TADCOBRB)

Address(es): MTU4.TADCOBRA 0008 8648h, MTU4.TADCOBRB 0008 864Ah

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Value after reset: | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

Note: • MTU4.TADCORA and MTU4.TADCORB must not be accessed in 8-bit units; they should be accessed in 16-bit units.

The TADCOBRA and TADCOBRB registers function as buffer registers for TADCORA and TADCORB, respectively. These registers specify the A/D converter start request cycle. When the crest or trough of the MTU4.TCNT count is reached, these register values are transferred to TADCORA and TADCORB, respectively.

## 20.2.12 Timer Counter (TCNT)

Address(es): MTU0.TCNT 0008 8706h, MTU1.TCNT 0008 8786h, MTU2.TCNT 0008 8806h, MTU3.TCNT 0008 8610h,MTU4.TCNT 0008 8612h, MTU5.TCNTU 0008 8880h, MTU5.TCNTV 0008 8890h, MTU5.TCNTW 0008 88A0h

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Note: • The TCNT counters must not be accessed in 8-bit units; they should be accessed in 16-bit units.

The MTU has a total of eight TCNT counters, one each for MTU0 to MTU4 and three (MTU5.TCNTU, TCNTV, and TCNTW) for MTU5. TCNT is a readable/writable counter.

## 20.2.13 Timer General Register (TGR)

Address(es): MTU0.TGRA 0008 8708h, MTU0.TGRB 0008 870Ah, MTU0.TGRC 0008 870Ch, MTU0.TGRD 0008 870Eh, MTU0.TGRE 0008 8720h, MTU0.TGRF 0008 8722h, MTU1.TGRA 0008 8788h, MTU1.TGRB 0008 878Ah, MTU2.TGRA 0008 8808h, MTU2.TGRB 0008 880Ah, MTU3.TGRA 0008 8618h, MTU3.TGRB 0008 861Ah, MTU3.TGRC 0008 8624h, MTU3.TGRD 0008 8626h, MTU4.TGRA 0008 861Ch, MTU4.TGRB 0008 861Eh, MTU4.TGRC 0008 8628h, MTU4.TGRD 0008 862Ah, MTU5.TGRU 0008 8882h, MTU5.TGRV 0008 8892h, MTU5.TGRW 0008 88A2h

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Value after reset: | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

Note: • The TGR registers must not be accessed in 8-bit units; they should be accessed in 16-bit units. TGR registers are initialized to FFFFh.

The MTU has a total of 21 TGR registers, six for MTU0, two each for MTU1 and MTU2, four each for MTU3 and MTU4, and three for MTU5.

TGRA, TGRB, TGRC, and TGRD function as either output compare or input capture registers. TGRC and TGRD for MTU0, MTU3, and MTU4 can also be designated for operation as buffer registers. TGR buffer register combinations are TGRA and TGRC, and TGRB and TGRD.

MTU0.TGRE and MTU0.TGRF function as compare registers. When the MTU0.TCNT count matches the MTU0.TGRE value, an A/D converter start request can be issued. TGRF can also be designated for operation as a buffer register. TGR buffer register combination is TGRE and TGRF.

MTU5.TGRU, MTU5.TGRV, and MTU5.TGRW function as compare match, input capture, or external pulse width measurement registers.

## 20.2.14 Timer Start Registers (TSTR)

- TSTR (MTU0 to MTU4)

Address(es): MTU.TSTR 0008 8680h

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | CST4 | CST3 | — | — | — | CST2 | CST1 | CST0 |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b0 | CST0 | Counter Start 0 | 0: MTU0.TCNT performs count stop<br>1: MTU0.TCNT performs count operation | R/W |
| b1 | CST1 | Counter Start 1 | 0: MTU1.TCNT performs count stop<br>1: MTU1.TCNT performs count operation | R/W |
| b2 | CST2 | Counter Start 2 | 0: MTU2.TCNT performs count stop<br>1: MTU2.TCNT performs count operation | R/W |
| b5 to b3 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |
| b6 | CST3 | Counter Start 3 | 0: MTU3.TCNT performs count stop<br>1: MTU3.TCNT performs count operation | R/W |
| b7 | CST4 | Counter Start 4 | 0: MTU4.TCNT performs count stop<br>1: MTU4.TCNT performs count operation | R/W |

The TSTR registers start or stop TCNT operation in MTU0 to MTU4.
Before setting the operating mode in TMDR or setting the TCNT count clock in TCR, be sure to stop the TCNT counter.

**CSTn Bits (Counter Start n) (n = 0 to 4)**

Each bit starts or stops TCNT in the corresponding channel.
If 0 is written to the CSTn bit during operation with the MTIOC pin designated for output, the counter stops but the output compare signal level from the MTIOC pin is retained. If TIOR is written to while the CSTn bit is 0, the pin output level will be changed to the specified initial output value.

- TSTR (MTU5)

Address(es): MTU5.TSTR 0008 88B4h

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | CSTU5 | CSTV5 | CSTW5 |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b0 | CSTW5 | Counter Start W5 | 0: MTU5.TCNTW count operation is stopped<br>1: MTU5.TCNTW performs count operation | R/W |
| b1 | CSTV5 | Counter Start V5 | 0: MTU5.TCNTV count operation is stopped<br>1: MTU5.TCNTV performs count operation | R/W |
| b2 | CSTU5 | Counter Start U5 | 0: MTU5.TCNTU count operation is stopped<br>1: MTU5.TCNTU performs count operation | R/W |
| b7 to b3 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |

## 20.2.15 Timer Synchronous Registers (TSYR)

Address(es): MTU.TSYR 0008 8681h

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | SYNC4 | SYNC3 | — | — | — | SYNC2 | SYNC1 | SYNC0 |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b0 | SYNC0 | Timer Synchronous Operation 0 | 0: MTU0.TCNT operates independently (TCNT presetting/clearing is not related to other channels).<br>1: MTU0.TCNT performs synchronous operation. TCNT synchronous presetting/synchronous clearing is enabled. | R/W |
| b1 | SYNC1 | Timer Synchronous Operation 1 | 0: MTU1.TCNT operates independently (TCNT presetting/clearing is not related to other channels).<br>1: MTU1.TCNT performs synchronous operation. TCNT synchronous presetting/synchronous clearing is enabled. | R/W |
| b2 | SYNC2 | Timer Synchronous Operation 2 | 0: MTU2.TCNT operates independently (TCNT presetting/clearing is not related to other channels).<br>1: MTU2.TCNT performs synchronous operation. TCNT synchronous presetting/synchronous clearing is enabled. | R/W |
| b5 to b3 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |
| b6 | SYNC3 | Timer Synchronous Operation 3 | 0: MTU3.TCNT operates independently (TCNT presetting/clearing is not related to other channels).<br>1: MTU3.TCNT performs synchronous operation. TCNT synchronous presetting/synchronous clearing is enabled. | R/W |
| b7 | SYNC4 | Timer Synchronous Operation 4 | 0: MTU4.TCNT operates independently (TCNT presetting/clearing is not related to other channels).<br>1: MTU4.TCNT performs synchronous operation. TCNT synchronous presetting/synchronous clearing is enabled. | R/W |

The TSYR registers select independent operation or synchronous operation of TCNT in MTU0 to MTU4.
A channel performs synchronous operation when the corresponding bit in TSYR is set to 1.

### SYNCn Bits (Timer Synchronous n Operation) (n = 0 to 4)

Each bit selects whether operation is independent of or synchronized with other channels.
When synchronous operation is selected, the TCNT synchronous presetting of multiple channels and synchronous clearing by counter clearing on another channel are possible.
To set synchronous operation, the SYNCn bits for at least two channels must be set to 1. To set synchronous clearing, in addition to the SYNCn bit, the TCNT clearing source must also be set by means of TCR.CCLR[2:0] bits.

## 20.2.16 Timer Read/Write Enable Registers (TRWER)

Address(es): MTU.TRWER 0008 8684h

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | RWE |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b0 | RWE | Read/Write Enable | 0: Read/write access to the registers is disabled<br>1: Read/write access to the registers is enabled | R/W |
| b7 to b1 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |

The TRWER registers enable or disable access to the registers and counters that have write-protection capability against accidental modification in MTU3 and MTU4.

### RWE Bit (Read/Write Enable)

This bit enables or disables access to the registers that have write-protection capability against accidental modification.
[Clearing condition]

- When 0 is written to the RWE bit after reading the RWE bit = 1

- Registers and Counters having Write-Protection Capability against Accidental Modification

22 registers: MTUn.TCR, MTUn.TMDR, MTUn.TIORH, MTUn.TIORL, MTUn.TIER, MTUn.TGRA, MTUn.TGRB, TOER, TOCR1, TOCR2, TGCR, TCDR, TDDR, and MTUn.TCNT (n = 3, 4)

## 20.2.17 Timer Output Master Enable Registers (TOER)

Address(es): MTU.TOER 0008 860Ah

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | OE4D | OE4C | OE3D | OE4B | OE4A | OE3B |
| Value after reset: | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b0 | OE3B | Master Enable MTIOC3B | 0: MTU output is disabled (inactive level)*1<br>1: MTU output is enabled | R/W |
| b1 | OE4A | Master Enable MTIOC4A | 0: MTU output is disabled (inactive level)*1<br>1: MTU output is enabled | R/W |
| b2 | OE4B | Master Enable MTIOC4B | 0: MTU output is disabled (inactive level)*1<br>1: MTU output is enabled | R/W |
| b3 | OE3D | Master Enable MTIOC3D | 0: MTU output is disabled (inactive level)*1<br>1: MTU output is enabled | R/W |
| b4 | OE4C | Master Enable MTIOC4C | 0: MTU output is disabled (inactive level)*1<br>1: MTU output is enabled | R/W |
| b5 | OE4D | Master Enable MTIOC4D | 0: MTU output is disabled (inactive level)*1<br>1: MTU output is enabled | R/W |
| b7, b6 | — | Reserved | These bits are read as 1. The write value should be 1. | R/W |

Note 1. The inactive level is determined by the settings in timer output control registers 1 and 2 (TOCR1 and TOCR2). For details, refer to section 20.2.18, Timer Output Control Registers 1 (TOCR1), and section 20.2.19, Timer Output Control Registers 2 (TOCR2). Set these bits to 1 to enable MTU output when complementary PWM or reset-synchronized PWM mode is not selected. When these bits are set to 0, the inactive level is output by setting in timer output control registers 1 and 2 (TOCR1 and TOCR).

The TOER registers enable or disable output settings for output pins MTIOC4D, MTIOC4C, MTIOC3D, MTIOC4B, MTIOC4A, and MTIOC3B.

These pins do not output correctly if the TOER bits have not been set. In MTU3 and MTU4, set TOER prior to setting TIOR.

Set TOER after clearing the CST3 and CST4 bits in TSTR to 0 (see Figure 20.35 and Figure 20.38).

## 20.2.18 Timer Output Control Registers 1 (TOCR1)

Address(es): MTU.TOCR 0008 860Eh

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | PSYE | — | — | TOCL | TOCS | OLSN | OLSP |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b0 | OLSP | Output Level Select P*2, *3 | See Table 20.30. | R/W |
| b1 | OLSN | Output Level Select N*2, *3 | See Table 20.31. | R/W |
| b2 | TOCS | TOC Select | 0: TOCR1 setting is selected<br>1: TOCR2 setting is selected | R/W |
| b3 | TOCL | TOC Register Write Protection*1 | 0: Write access to the TOCS, OLSN, and OLSP bits is enabled<br>1: Write access to the TOCS, OLSN, and OLSP bits is disabled | R/W*4 |
| b5, b4 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |
| b6 | PSYE | PWM Synchronous Output Enable | 0: Toggle output is disabled<br>1: Toggle output is enabled | R/W |
| b7 | — | Reserved | This bit is read as 0. The write value should be 0. | R/W |

Note 1. Setting the TOCR1.TOCL bit to 1 prevents accidental modification when the CPU goes out of control.
Note 2. Clearing the TOCR1.TOCS bit to 0 makes this bit setting valid.
Note 3. If dead-time is not generated, the negative-phase output is always the exact inverse of the positive-phase output. In this case, only the OLSP bit is valid.
Note 4. This bit can be set to 1 only once after a power-on reset. After 1 is written, 0 cannot be written to the bit.

The TOCR1 registers enable or disable PWM-synchronized toggle output in complementary PWM mode and reset-synchronized PWM mode, and control inversion of PWM output level.

### OLSP Bit (Output Level Select P)

This bit selects the positive-phase output level in reset-synchronized PWM mode and complementary PWM mode.

### OLSN Bit (Output Level Select N)

This bit selects the negative-phase output level in reset-synchronized PWM mode and complementary PWM mode.

### TOCS Bit (TOC Select)

This bit selects either the TOCR1 or TOCR2 setting to be used for the output level in complementary PWM mode and reset-synchronized PWM mode.

### TOCL Bit (TOC Register Write Protection)

This bit enables or disables write access to the TOCS, OLSN, and OLSP bits in TOCR1.

### PSYE Bit (PWM Synchronous Output Enable)

This bit enables or disables toggle output synchronized with the PWM cycle.

**Table 20.30 Output Level Select Function**

| Bit 0 | Function | | | |
|---|---|---|---|---|
| | | | Compare Match Output | |
| OLSP | Initial Output | Active Level | Up-Counting | Down-Counting |
| 0 | High | Low | Low | High |
| 1 | Low | High | High | Low |

**Table 20.31 Output Level Select Function**

| Bit 1 | Function | | | |
|---|---|---|---|---|
| | | | Compare Match Output | |
| OLSN | Initial Output | Active Level | Up-Counting | Down-Counting |
| 0 | High | Low | High | Low |
| 1 | Low | High | Low | High |

Note: • The initial output value of negative-phase waveform changes to an active level after the dead time has passed since counting starts.

Figure 20.2 shows an example of output in complementary PWM mode (one phase) when OLSN = 1 and OLSP = 1.

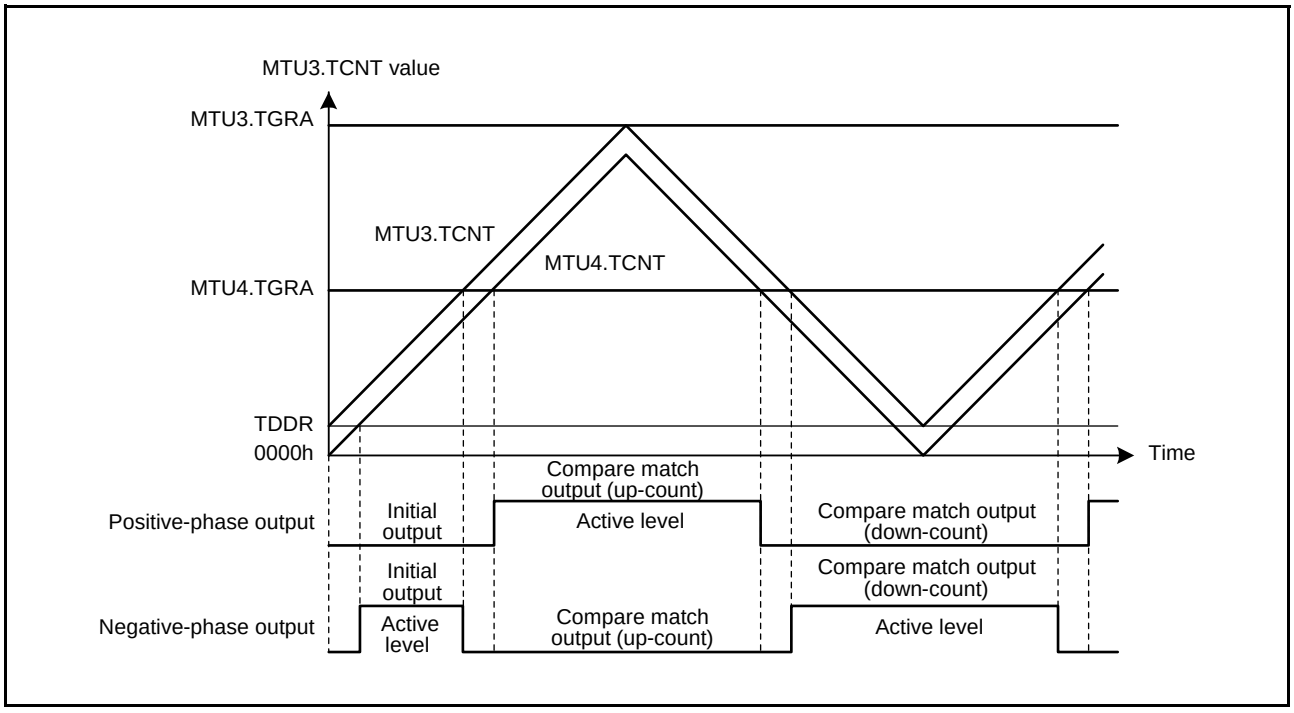



**Figure 20.2 Example of Output in Complementary PWM Mode**

## 20.2.19 Timer Output Control Registers 2 (TOCR2)

Address(es): MTU.TOCR2 0008 860Fh

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | BF[1:0] | | OLS3N | OLS3P | OLS2N | OLS2P | OLS1N | OLS1P |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b0 | OLS1P | Output Level Select 1P*1, *2 | This bit selects the output level on MTIOC3B in reset-synchronized PWM mode and complementary PWM mode. See Table 20.32. | R/W |
| b1 | OLS1N | Output Level Select 1N*1, *2 | This bit selects the output level on MTIOC3D in reset-synchronized PWM mode and complementary PWM mode. See Table 20.33. | R/W |
| b2 | OLS2P | Output Level Select 2P*1, *2 | This bit selects the output level on MTIOC4A in reset-synchronized PWM mode and complementary PWM mode. See Table 20.34. | R/W |
| b3 | OLS2N | Output Level Select 2N*1, *2 | This bit selects the output level on MTIOC4C in reset-synchronized PWM mode and complementary PWM mode. See Table 20.35. | R/W |
| b4 | OLS3P | Output Level Select 3P*1, *2 | This bit selects the output level on MTIOC4B in reset-synchronized PWM mode and complementary PWM mode. See Table 20.36. | R/W |
| b5 | OLS3N | Output Level Select 3N*1, *2 | This bit selects the output level on MTIOC4D in reset-synchronized PWM mode and complementary PWM mode. See Table 20.37. | R/W |
| b7, b6 | BF[1:0] | TOLBR Buffer Transfer Timing Select | These bits select the timing for transferring data from TOLBR to TOCR2. See Table 20.38 for details. | R/W |

Note 1. Setting the TOCR1.TOCS bit to 1 makes this bit setting valid.
Note 2. If dead-time is not generated, the negative-phase output is always the exact inverse of the positive-phase output. In these cases, only the OLSiP bits are valid (i = 1 to 3).

The TOCR2 registers control inversion of PWM output level in complementary PWM mode and reset-synchronized PWM mode.

**Table 20.32 MTIOC3B Output Level Select Function**

| Bit 0 | Function | | | |
|---|---|---|---|---|
| | | | Compare Match Output | |
| OLS1P | Initial Output | Active Level | Up-Counting | Down-Counting |
| 0 | High | Low | Low | High |
| 1 | Low | High | High | Low |

**Table 20.33 MTIOC3D Output Level Select Function**

| Bit 1 | Function | | | |
|---|---|---|---|---|
| | | | Compare Match Output | |
| OLS1N | Initial Output | Active Level | Up-Counting | Down-Counting |
| 0 | High | Low | High | Low |
| 1 | Low | High | Low | High |

Note: • The initial output value of negative-phase waveform changes to an active level after the dead time has passed since counting starts.

**Table 20.34 MTIOC4A Output Level Select Function**

| Bit 2 | Function | | | |
|---|---|---|---|---|
| | | | Compare Match Output | |
| OLS2P | Initial Output | Active Level | Up-Counting | Down-Counting |
| 0 | High | Low | Low | High |
| 1 | Low | High | High | Low |

**Table 20.35 MTIOC4C Output Level Select Function**

| Bit 3 | Function | | | |
|---|---|---|---|---|
| | | | Compare Match Output | |
| OLS2N | Initial Output | Active Level | Up-Counting | Down-Counting |
| 0 | High | Low | High | Low |
| 1 | Low | High | Low | High |

Note: • The initial output value of negative-phase waveform changes to an active level after the dead time has passed since counting starts.

**Table 20.36 MTIOC4B Output Level Select Function**

| Bit 4 | Function | | | |
|---|---|---|---|---|
| | | | Compare Match Output | |
| OLS3P | Initial Output | Active Level | Up-Counting | Down-Counting |
| 0 | High | Low | Low | High |
| 1 | Low | High | High | Low |

**Table 20.37 MTIOC4D Output Level Select Function**

| Bit 5 | Function | | | |
|---|---|---|---|---|
| | | | Compare Match Output | |
| OLS3N | Initial Output | Active Level | Up-Counting | Down-Counting |
| 0 | High | Low | High | Low |
| 1 | Low | High | Low | High |

Note: • The initial output value of negative-phase waveform changes to an active level after the dead time has passed since counting starts.

**Table 20.38 Setting of TOCR2.BF[1:0] Bits**

| Bit 7 | Bit 6 | Description | |
|---|---|---|---|
| BF1 | BF0 | Complementary PWM Mode | Reset-Synchronized PWM Mode |
| 0 | 0 | Does not transfer data from the buffer register (TOLBR) to TOCR2. | Does not transfer data from the buffer register (TOLBR) to TOCR2. |
| 0 | 1 | Transfers data from the buffer register (TOLBR) to TOCR2 at the crest of the MTU4.TCNT count. | Transfers data from the buffer register (TOLBR) to TOCR2 when MTU4.TCNT or MTU3.TCNT is cleared. |
| 1 | 0 | Transfers data from the buffer register (TOLBRj) to TOCR2 at the trough of the MTU4.TCNT count. | Setting prohibited |
| 1 | 1 | Transfers data from the buffer register (TOLBR) to TOCR2 at the crest and trough of the MTU4.TCNT count. | Setting prohibited |

## 20.2.20 Timer Output Level Buffer Registers (TOLBR)

Address(es): MTU.TOLBR 0008 8636h

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | OLS3N | OLS3P | OLS2N | OLS2P | OLS1N | OLS1P |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b0 | OLS1P | Output Level Select 1P | Specify the buffer value to be transferred to the OLS1P bit in TOCR2. | R/W |
| b1 | OLS1N | Output Level Select 1N | Specify the buffer value to be transferred to the OLS1N bit in TOCR2. | R/W |
| b2 | OLS2P | Output Level Select 2P | Specify the buffer value to be transferred to the OLS2P bit in TOCR2. | R/W |
| b3 | OLS2N | Output Level Select 2N | Specify the buffer value to be transferred to the OLS2N bit in TOCR2. | R/W |
| b4 | OLS3P | Output Level Select 3P | Specify the buffer value to be transferred to the OLS3P bit in TOCR2. | R/W |
| b5 | OLS3N | Output Level Select 3N | Specify the buffer value to be transferred to the OLS3N bit in TOCR2. | R/W |
| b7, b6 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |

The TOLBR registers function as buffer registers for TOCR2 and specify the PWM output level in complementary PWM mode and reset-synchronized PWM mode.

Figure 20.3 shows an example of the PWM output level setting procedure in buffer operation.

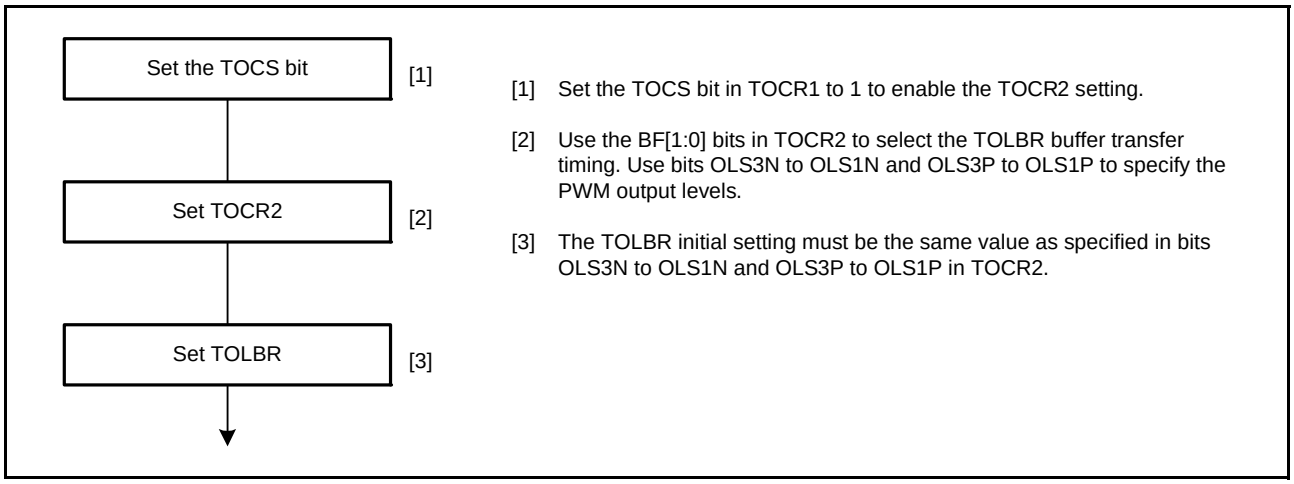



**Figure 20.3 Example of PWM Output Level Setting Procedure in Buffer Operation**

## 20.2.21 Timer Gate Control Registers (TGCR)

Address(es): MTU.TGCR 0008 860Dh

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | BDC | N | P | FB | WF | VF | UF |
| Value after reset: | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b0 | UF | Output Phase Switch | These bits turn on or off the positive-phase/negative-phase output. The setting of these bits is valid only when the TGCR.FB bit is set to 1. In this case, the setting of b0 to b2 is used instead of the external input. See Table 20.39. | R/W |
| b1 | VF | | | R/W |
| b2 | WF | | | R/W |
| b3 | FB | External Feedback Signal Enable | 0: Output is switched by external input (input sources are TGRA, TGRB, and TGRC input capture signals in MTU0)<br>1: Output is switched by software (TGCR's UF, VF, and WF settings) | R/W |
| b4 | P | Positive-Phase Output (P) Control | 0: Level output<br>1: Reset-synchronized PWM or complementary PWM output | R/W |
| b5 | N | Negative-Phase Output (N) Control | 0: Level output<br>1: Reset-synchronized PWM or complementary PWM output | R/W |
| b6 | BDC | Brushless DC Motor | 0: Ordinary output<br>1: Functions of this register are made effective | R/W |
| b7 | — | Reserved | This bit is read as 1. The write value should be 1. | R/W |

The TGCR registers control the output waveform necessary for brushless DC motor control in reset-synchronized PWM mode and complementary PWM mode. These register settings are ineffective for anything other than complementary PWM mode and reset-synchronized PWM mode.

### UF, VF, and WF Bits (Output Phase Switch)

The setting of these bits is valid only when the TGCR.FB bit is set to 1. In this case, the setting of b0 to b2 is used instead of the external input. See Table 20.39.

### FB Bit (External Feedback Signal Enable)

This bit selects whether the positive-/negative-phase output is switched automatically with the TGRA, TGRB, and TGRC input capture signals in MTU0 or by writing 0 or 1 to bits 2 to 0 in TGCR.

### P Bit (Positive-Phase Output (P) Control)

This bit selects the level output or the reset-synchronized PWM/complementary PWM output for the positive-phase output pins (MTIOC3B, MTIOC4A, and MTIOC4B pins).

### N Bit (Negative-Phase Output (N) Control)

This bit selects the level output or the reset-synchronized PWM/complementary PWM output for the negative-phase output pins (MTIOC3D, MTIOC4C, and MTIOC4D pins).

### BDC Bit (Brushless DC Motor)

This bit selects whether to make the functions of TGCR effective or ineffective.

**Table 20.39 Output Level Select Function**

| Bit 2 | Bit 1 | Bit 0 | Function | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | | MTIOC3B | MTIOC4A | MTIOC4B | MTIOC3D | MTIOC4C | MTIOC4D |
| WF | VF | UF | U Phase | V Phase | W Phase | U Phase | V Phase | W Phase |
| 0 | 0 | 0 | OFF | OFF | OFF | OFF | OFF | OFF |
| 0 | 0 | 1 | ON | OFF | OFF | OFF | OFF | ON |
| 0 | 1 | 0 | OFF | ON | OFF | ON | OFF | OFF |
| 0 | 1 | 1 | OFF | ON | OFF | OFF | OFF | ON |
| 1 | 0 | 0 | OFF | OFF | ON | OFF | ON | OFF |
| 1 | 0 | 1 | ON | OFF | OFF | OFF | ON | OFF |
| 1 | 1 | 0 | OFF | OFF | ON | ON | OFF | OFF |
| 1 | 1 | 1 | OFF | OFF | OFF | OFF | OFF | OFF |

## 20.2.22 Timer Subcounters (TCNTS)

Address(es): MTU.TCNTS 0008 8620h

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Note: • The TCNTS counters must not be accessed in 8-bit units; they should be accessed in 16-bit units.

The TCNTS counters are read-only counters that are used only in complementary PWM mode.

## 20.2.23 Timer Dead Time Data Registers (TDDR)

Address(es): MTU.TDDR 0008 8616h

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Value after reset: | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

Note: • The TDDR registers must not be accessed in 8-bit units; they should be accessed in 16-bit units.

The TDDR registers specify the MTU3.TCNT and MTU4.TCNT counter offset value in complementary PWM mode. In complementary PWM mode, when the MTU3.TCNT and MTU4.TCNT counters are cleared and then restarted, the TDDR value is loaded into the MTU3.TCNT counter and the count operation starts.

## 20.2.24 Timer Cycle Data Registers (TCDR)

Address(es): MTU.TCDR 0008 8614h

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Value after reset: | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

Note: • The TCDR registers must not be accessed in 8-bit units; they should be accessed in 16-bit units.

The TCDR registers specify the count value to switch the count direction of the TCNTS counter. These registers are used only in complementary PWM mode. Set half the PWM carrier cycle as the TCDR value. TCDR is constantly compared with the TCNTS counter in complementary PWM mode, and when a match occurs, the TCNTS counter switches direction (down-count to up-count).

## 20.2.25 Timer Cycle Buffer Registers (TCBR)

Address(es): MTU.TCBR 0008 8622h

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Value after reset: | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

Note: • The TCBR registers must not be accessed in 8-bit units; they should be accessed in 16-bit units.

The TCBR registers function as buffer registers for TCDR, and specify the count value to switch the count direction of the TCNTS counter. These registers are used only in complementary PWM mode. The TCBR value is transferred to TCDR with the transfer timing set in TMDR.

## 20.2.26 Timer Interrupt Skipping Set Registers (TITCR)

Address(es): MTU.TITCR 0008 8630h

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | T3AEN | T3ACOR[2:0] | | | T4VEN | T4VCOR[2:0] | | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b2 to b0 | T4VCOR[2:0] | TCIV4 Interrupt Skipping Count Setting | These bits specify the TCIV4 interrupt skipping count within the range from 0 to 7.*1 For details, see Table 20.40. | R/W |
| b3 | T4VEN | T4VEN | 0: TCIV4 interrupt skipping disabled<br>1: TCIV4 interrupt skipping enabled | R/W |
| b6 to b4 | T3ACOR[2:0] | TGIA3 Interrupt Skipping Count Setting | These bits specify the TGIA3 interrupt skipping count within the range from 0 to 7.*1 For details, see Table 20.41. | R/W |
| b7 | T3AEN | T3AEN | 0: TGIA3 interrupt skipping disabled<br>1: TGIA3 interrupt skipping enabled | R/W |

Note 1. When 0 is specified for the interrupt skipping count, no interrupt skipping will be performed.
Before changing the interrupt skipping count, be sure to clear the TITCR.T3AEN and TITCR.T4VEN bits to 0 to clear the timer interrupt skipping counter (TITCNT).

**Table 20.40 Setting of Interrupt Skipping Count by T4VCOR[2:0] Bits**

| Bit 2 | Bit 1 | Bit 0 | |
|---|---|---|---|
| T4VCOR2 | T4VCOR1 | T4VCOR0 | Description |
| 0 | 0 | 0 | Does not perform TCIV4 interrupt skipping. |
| 0 | 0 | 1 | Sets the TCIV4 interrupt skipping count to 1. |
| 0 | 1 | 0 | Sets the TCIV4 interrupt skipping count to 2. |
| 0 | 1 | 1 | Sets the TCIV4 interrupt skipping count to 3. |
| 1 | 0 | 0 | Sets the TCIV4 interrupt skipping count to 4. |
| 1 | 0 | 1 | Sets the TCIV4 interrupt skipping count to 5. |
| 1 | 1 | 0 | Sets the TCIV4 interrupt skipping count to 6. |
| 1 | 1 | 1 | Sets the TCIV4 interrupt skipping count to 7. |

**Table 20.41 Setting of Interrupt Skipping Count by T3ACOR[2:0] Bits**

| Bit 6 | Bit 5 | Bit 4 | |
|---|---|---|---|
| T3ACOR2 | T3ACOR1 | T3ACOR0 | Description |
| 0 | 0 | 0 | Does not perform TGIA3 interrupt skipping. |
| 0 | 0 | 1 | Sets the TGIA3 interrupt skipping count to 1. |
| 0 | 1 | 0 | Sets the TGIA3 interrupt skipping count to 2. |
| 0 | 1 | 1 | Sets the TGIA3 interrupt skipping count to 3. |
| 1 | 0 | 0 | Sets the TGIA3 interrupt skipping count to 4. |
| 1 | 0 | 1 | Sets the TGIA3 interrupt skipping count to 5. |
| 1 | 1 | 0 | Sets the TGIA3 interrupt skipping count to 6. |
| 1 | 1 | 1 | Sets the TGIA3 interrupt skipping count to 7. |

## 20.2.27 Timer Interrupt Skipping Counters (TITCNT)

Address(es): MTU.TITCNT 0008 8631h

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | T3ACNT[2:0] | | | — | T4VCNT[2:0] | | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b2 to b0 | T4VCNT[2:0] | TCIV4 Interrupt Counter | While the T4VEN bit in TITCR is set to 1, the count in these bits is incremented every time a TCIV4 interrupt source occurs. | R |
| b3 | — | Reserved | This bit is read as 0. Writing to this bit has no effect. | R |
| b6 to b4 | T3ACNT[2:0] | TGIA3 Interrupt Counter | While the T3AEN bit in TITCR is set to 1, the count in these bits is incremented every time a TGIA3 interrupt source occurs. | R |
| b7 | — | Reserved | This bit is read as 0. Writing to this bit has no effect. | R |

Note: • To clear the TITCNT, clear the T3AEN and T4VEN bits in TITCR to 0.

The TITCNT counters count the number of interrupt source occurrences for interrupt skipping. TITCNT retain their values even after stopping the count operation of MTU4.TCNT and MTU3.TCNT.

**T4VCNT[2:0] Bits (TCIV4 Interrupt Counter)**

[Clearing conditions]

- When the T4VCNT[2:0] bits in TITCNT match the T4VCOR[2:0] bits in TITCR
- When the T4VEN bit in TITCR is cleared to 0
- When the T4VCOR[2:0] bits in TITCR are cleared to 000b

**T3ACNT[2:0] Bits (TGIA3 Interrupt Counter)**

[Clearing conditions]

- When the T3ACNT[2:0] bits in TITCNT match the T3ACOR[2:0] bits in TITCR
- When the T3AEN bit in TITCR is cleared to 0
- When the T3ACOR[2:0] bits in TITCR are cleared to 000b

## 20.2.28 Timer Buffer Transfer Set Registers (TBTER)

Address(es): MTU.TBTER 0008 8632h

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | BTE[1:0] | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b1, b0 | BTE[1:0] | Buffer Transfer Disable and Interrupt Skipping Link Setting | These bits enable or disable transfer from the buffer registers used in complementary PWM mode to the temporary registers and specify whether to link the transfer with interrupt skipping operation. See Table 20.42 for details. | R/W |
| b7 to b2 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |

The TBTER registers enable or disable transfer from the buffer registers used in complementary PWM mode to the temporary registers and specify whether to link the transfer with interrupt skipping operation.

**Table 20.42 Setting of TBTER.BTE[1:0] Bits**

| Bit 1 | Bit 0 | |
|---|---|---|
| BTE1 | BTE0 | Description |
| 0 | 0 | Enables transfer from the buffer registers to the temporary registers*1 and does not link the transfer with interrupt skipping operation. |
| 0 | 1 | Disables transfer from the buffer registers to the temporary registers. |
| 1 | 0 | Links transfer from the buffer registers to the temporary registers with interrupt skipping operation.*2 |
| 1 | 1 | Setting prohibited |

Note: • Target buffer registers: MTU3.TGRC, MTU3.TGRD, MTU4.TGRC, MTU4.TGRD, and MTU.TCBRA

Note 1. Data is transferred in accordance with the TMDT.MD[3:0] bit setting. For details, refer to section 20.3.8, Complementary PWM Mode.

Note 2. When interrupt skipping is disabled (the T3AEN and T4VEN bits are cleared to 0 in the timer interrupt skipping set register (TITCR) or the interrupt skipping count setting bits (T3ACOR and T4VCOR) in TITCR are cleared to 0), be sure to disable link of buffer transfer with interrupt skipping (clear the BTE1 bit in the timer buffer transfer set register (TBTER) to 0). If link with interrupt skipping is enabled while interrupt skipping is disabled, buffer transfer will not be performed.

## 20.2.29 Timer Dead Time Enable Registers (TDER)

Address(es): MTU.TDER 0008 8634h

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | TDER |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b0 | TDER | Dead Time Enable | 0: No dead time is generated<br>1: Dead time is generated*1 | R/(W) |
| b7 to b1 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |

Note 1. TDDR must be set to 1 or a larger value.

The TDER registers specify dead time generation in complementary PWM mode. The MTU3 has one TDER register. TDER should be modified only while TCNT stops.

### TDER Bit (Dead Time Enable)

This bit specifies whether to generate dead time.
[Clearing condition]

- When 0 is written to the TDER bit after reading the TDER bit = 1

## 20.2.30 Timer Waveform Control Registers (TWCR)

Address(es): MTU.TWCR 0008 8660h

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | CCE | — | — | — | — | — | — | WRE |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b0 | WRE | Initial Output Inhibition Enable | 0: Initial value specified in TOCR is output<br>1: Initial output is inhibited | R/(W) *1 |
| b6 to b1 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |
| b7 | CCE | Compare Match Clear Enable | 0: Counters are not cleared at MTU3.TGRA compare match<br>1: Counters are cleared at MTU3.TGRA compare match | R/(W) *2 |

Note 1. Do not set this bit to 1 unless complementary PWM mode is selected.
Note 2. Do not set this bit to 1 unless complementary PWM mode 1 is selected.

The TWCR registers control the output waveform when synchronous counter clearing occurs in MTU3.TNCT and MTU4.TNCT in complementary PWM mode and specifies whether to clear the counters at MTU3.TGRA compare match.
The TWCR.CCE bit and TWCR.WRE bit should be modified only while TCNT stops.

### WRE Bit (Initial Output Inhibition Enable)

This bit selects the waveform output when synchronous counter clearing occurs in complementary PWM mode.
The initial output is prohibited only when synchronous clearing occurs within the Tb interval at the trough in complementary PWM mode. When synchronous clearing occurs outside this interval, the initial value specified in TOCR is output regardless of the WRE bit setting. The initial value specified in TOCR is also output when synchronous clearing occurs in the Tb interval at the trough immediately after MTU3.TCNT and MTU4.TCNT start operation.
For the Tb interval at the trough in complementary PWM mode, see Figure 20.40.
[Setting condition]

- When 1 is written to the WRE bit after reading the WRE bit = 0

### CCE Bit (Compare Match Clear Enable)

This bit specifies whether to clear counters at TGRA3 compare match in complementary PWM mode 1.
[Setting condition]

- When 1 is written to the CCE bit after reading the CCE bit = 0

## 20.2.31 Noise Filter Control Registers (NFCR)

- NFCR (MTU0 to MTU4)

Address(es): MTU0.NFCR 0008 8690h, MTU1.NFCR 0008 8691h, MTU2.NFCR 0008 8692h, MTU3.NFCR 0008 8693h, MTU4.NFCR 0008 8694h

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | NFCS[1:0] | | NFDEN | NFCEN | NFBEN | NFAEN |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b0 | NFAEN | Noise Filter A Enable | 0: The noise filter for the MTIOCnA pin is disabled.<br>1: The noise filter for the MTIOCnA pin is enabled. | R/W |
| b1 | NFBEN | Noise Filter B Enable | 0: The noise filter for the MTIOCnB pin is disabled.<br>1: The noise filter for the MTIOCnB pin is enabled. | R/W |
| b2 | NFCEN | Noise Filter C Enable | 0: The noise filter for the MTIOCnC pin is disabled.<br>1: The noise filter for the MTIOCnC pin is enabled. | R/W*1 |
| b3 | NFDEN | Noise Filter D Enable | 0: The noise filter for the MTIOCnD pin is disabled.<br>1: The noise filter for the MTIOCnD pin is enabled. | R/W*1 |
| b5, b4 | NFCS[1:0] | Noise Filter Clock Select | b5 b4<br>0 0: PCLK/1<br>0 1: PCLK/8<br>1 0: PCLK/32<br>1 1: The clock source for counting is the external clock. | R/W |
| b7, b6 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |

Note 1. These bits are reserved in the NFCR for MTU1 and MTU2. These bits are read as 0, and writing to them is not possible.

The MTUn.NFCR registers (n = 0 to 4) enable and disable the noise filters for the MTIOCnm (n = 0 to 4; m = A to D) pins and sets the sampling clocks for the noise filters.

**NFAEN Bit (Noise Filter A Enable)**

This bit disables or enables the noise filter for input from the MTIOCnA pin. Since unexpected edges may be internally generated when the value of NFAEN is changed, select the output compare function in the timer I/O control register and set the TMDR.MD[3:0] bits to a value other than that for normal mode (0000b) before changing the value.

**NFBEN Bit (Noise Filter B Enable)**

This bit disables or enables the noise filter for input from the MTIOCnB pin. Since unexpected edges may be internally generated when the value of NFBEN is changed, select the output compare function in the timer I/O control register and set the TMDR.MD[3:0] bits to a value other than that for normal mode (0000b) before changing the value.

**NFCEN Bit (Noise Filter C Enable)**

This bit disables or enables the noise filter for input from the MTIOCnC pin. Since unexpected edges may be internally generated when the value of NFCEN is changed, select the output compare function in the timer I/O control register and set the TMDR.MD[3:0] bits to a value other than that for normal mode (0000b) before changing the value.

**NFDEN Bit (Noise Filter D Enable)**

This bit disables or enables the noise filter for input from the MTIOCnD pin. Since unexpected edges may be internally generated when the value of NFDEN is changed, select the output compare function in the timer I/O control register and set the TMDR.MD[3:0] bits to a value other than that for normal mode (0000b) before changing the value.

**NFCS[1:0] Bits (Noise Filter Clock Select)**

These bits set the sampling interval for the noise filters. When setting the NFCS[1:0] bits, wait for two cycles of the selected sampling interval before setting the input-capture function. When the NFCS[1:0] bits are set to 11b, selecting the external clock as the source to drive counting, wait for two cycles of the external clock before setting the input-capture function.

- NFCR (MTU5)

Address(es): MTU5.NFCR 0008 8695h

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | NFCS[1:0] | | — | NFWEN | NFVEN | NFUEN |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b0 | NFUEN | Noise Filter U Enable | 0: The noise filter for the MTIC5U pin is disabled.<br>1: The noise filter for the MTIC5U pin is enabled. | R/W |
| b1 | NFVEN | Noise Filter V Enable | 0: The noise filter for the MTIC5V pin is disabled.<br>1: The noise filter for the MTIC5V pin is enabled. | R/W |
| b2 | NFWEN | Noise Filter W Enable | 0: The noise filter for the MTIC5W pin is disabled.<br>1: The noise filter for the MTIC5W pin is enabled. | R/W |
| b3 | — | Reserved | This bit is read as 0. The write value should be 0. | R/W |
| b5, b4 | NFCS[1:0] | Noise Filter Clock Select | b5 b4<br>0 0: PCLK/1<br>0 1: PCLK/8<br>1 0: PCLK/32<br>1 1: The clock source for counting is the external clock. | R/W |
| b7, b6 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |

MTU5.NFCR is 8-bit readable and writable register. This register controls enabling and disabling of the noise filters for the MTIC5m (m = U, V, W) pins and sets the sampling clock for the noise filters.

### NFUEN Bit (Noise Filter U Enable)

This bit disables or enables the noise filter for input from the MTIC5U pin. Since unexpected edges may be internally generated when the value of NFUEN is changed, select the compare-match function in the timer I/O control register before changing the value.

### NFVEN Bit (Noise Filter V Enable)

This bit disables or enables the noise filter for input from the MTIC5V pin. Since unexpected edges may be internally generated when the value of NFVEN is changed, select the compare-match function in the timer I/O control register before changing the value.

### NFWEN Bit (Noise Filter W Enable)

This bit disables or enables the noise filter for input from the MTIC5W pin. Since unexpected edges may be internally generated when the value of NFWEN is changed, select the compare-match function in the timer I/O control register before changing the value.

### NFCS[1:0] Bits (Noise Filter Clock Select)

These bits set the sampling interval for the noise filters. When setting the NFCS[1:0] bits, wait for two cycles of the selected sampling interval before setting the input-capture function.

### 20.2.32 Bus Master Interface

The timer counters (TCNT), timer general registers (TGR), timer subcounter (TCNTS), timer cycle buffer register (TCBR), timer dead time data register (TDDR), timer cycle data register (TCDR), timer A/D converter start request control register (TADCR), timer A/D converter start request cycle set registers (TADCORA/TADCORB), and timer A/D converter start request cycle set buffer registers (TADCORA/TADCORB) are 16-bit registers. A 16-bit data bus to the bus master enables 16-bit read/write access. 8-bit read/write is not allowed. Access the registers in 16-bit units.
All registers other than the above registers are 8-bit registers, so read/write access should be performed in 8-bit units.

## 20.3 Operation

### 20.3.1 Basic Functions

Each channel has TCNT and TGR. TCNT performs up-counting, and is also capable of free-running operation, periodic counting, and external event counting.
Each TGR can be used as an input capture register or an output compare register.

#### (1) Counter Operation

When one of bits CST0 to CST4 in TSTR or bits CSTU5, CSTV5, and CSTW5 in MTU5.TSTR is set to 1, TCNT for the corresponding channel begins counting. TCNT can operate as a free-running counter, periodic counter, for example.

#### (a) Example of Count Operation Setting Procedure

Figure 20.4 shows an example of the count operation setting procedure.

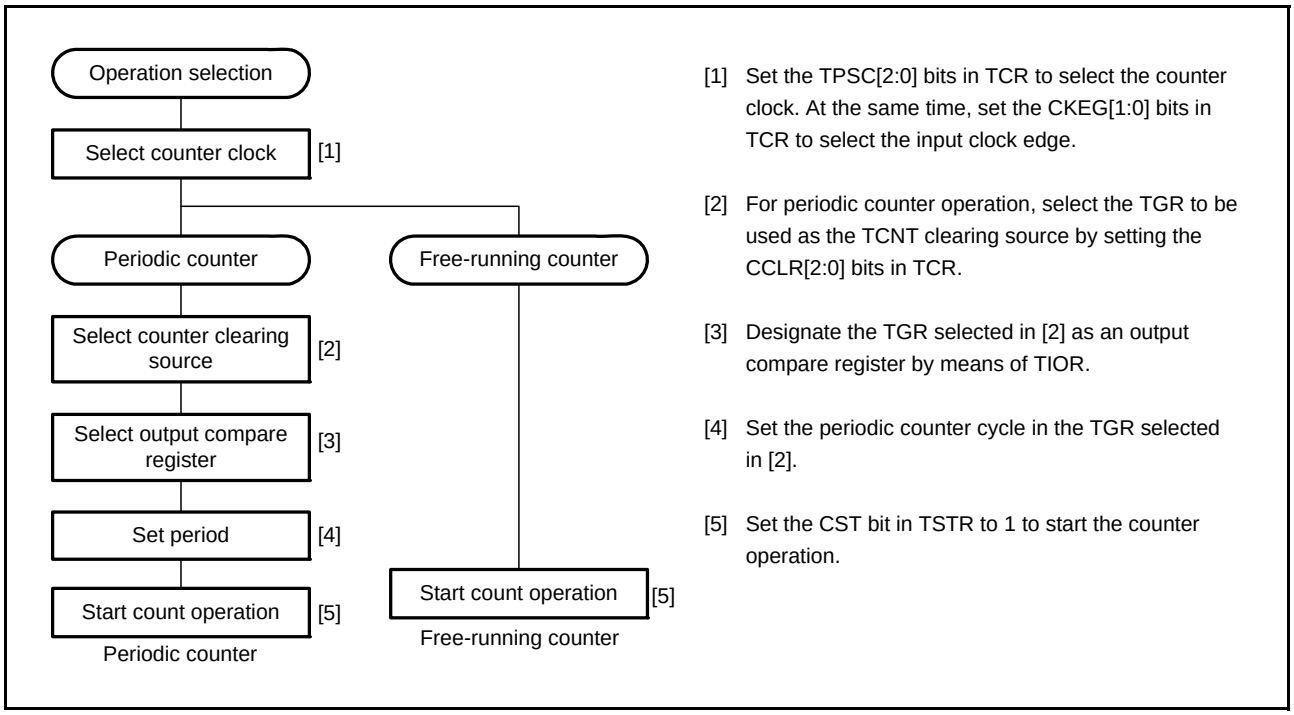



**Figure 20.4 Example of Counter Operation Setting Procedure**

### (b) Free-Running Count Operation and Periodic Count Operation

Immediately after a reset, the MTU's TCNT counters are all designated as free-running counters. When the relevant CSTn bit in TSTR is set to 1, the corresponding TCNT counter starts up-count operation as a free-running counter. When TCNT overflows (from FFFFh to 0000h), the MTU requests an interrupt if the corresponding TCIEV bit in TIER is 1. After an overflow, TCNT starts counting up again from 0000h.

Figure 20.5 illustrates free-running counter operation.

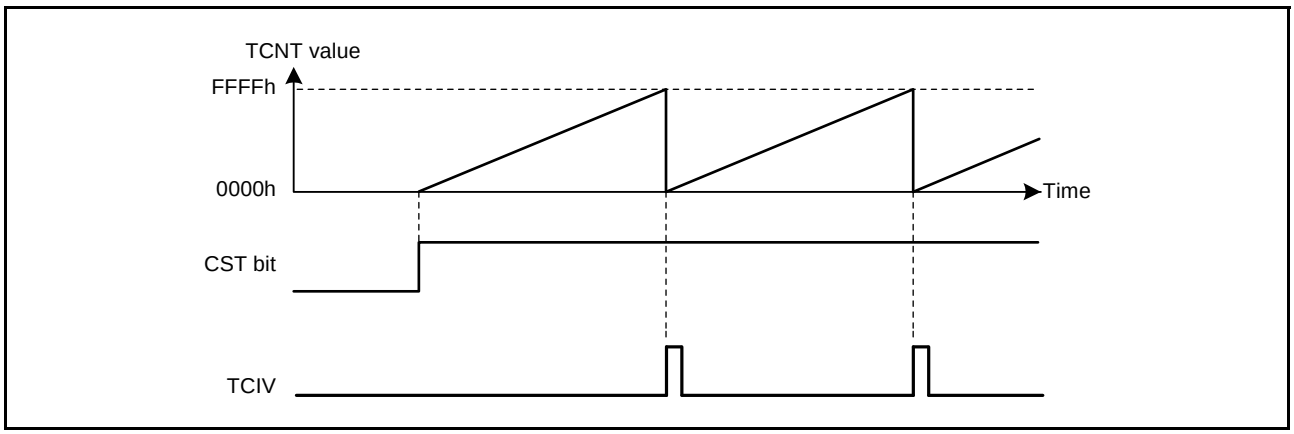



**Figure 20.5 Free-Running Counter Operation**

When compare match is selected as the TCNT clearing source, TCNT for the relevant channel performs periodic count operation. TGR for setting the cycle is designated as an output compare register, and counter clearing by compare match is selected by means of bits CCLR[2:0] in TCR. After the settings have been made, TCNT starts up-count operation as a periodic counter when the corresponding bit in TSTR is set to 1. When the count matches the value in TGR, TCNT is cleared to 0000h.

If the value of the corresponding TGIE bit in TIER is 1 at this point, the MTU requests an interrupt. After a compare match, TCNT starts counting up again from 0000h.

Figure 20.6 illustrates periodic counter operation.

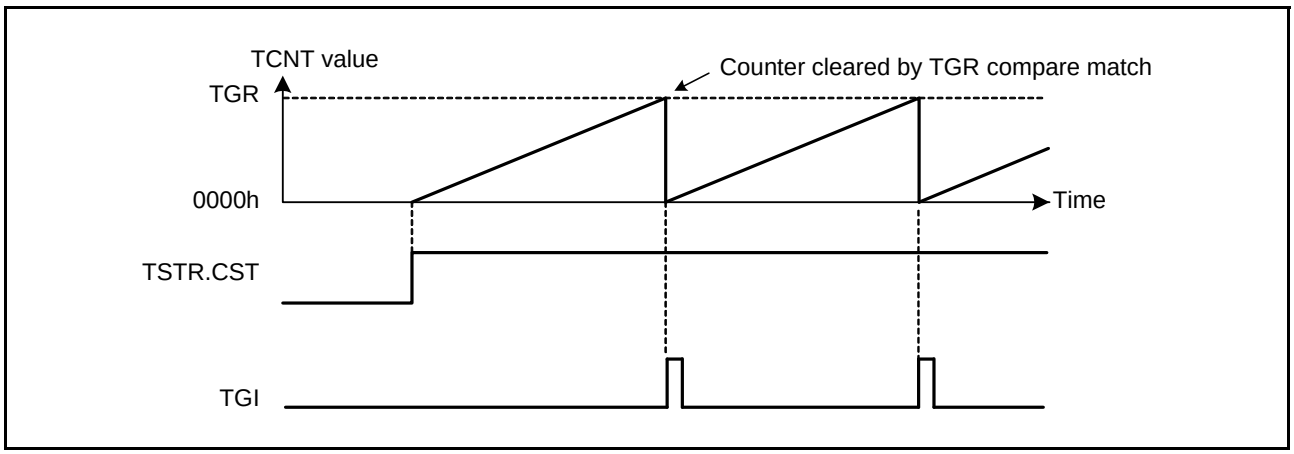



**Figure 20.6 Periodic Counter Operation**

### (2) Waveform Output by Compare Match

The MTU can output low or high or toggle output from the corresponding output pin using compare match.

#### (a) Example of Procedure for Setting Waveform Output by Compare Match

Figure 20.7 shows an example of the procedure for setting waveform output by compare match.

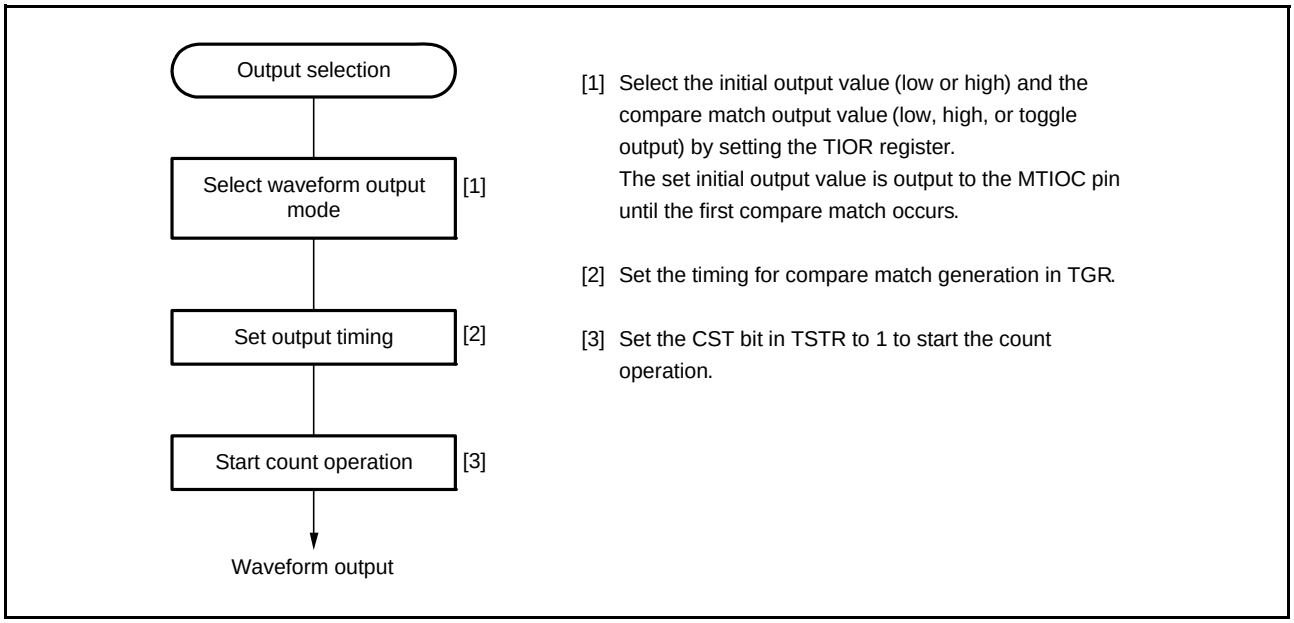



**Figure 20.7 Example of Procedure for Setting Waveform Output by Compare Match**

### (b) Examples of Waveform Output Operation

Figure 20.8 shows an example of low output and high output.
In this example, TCNT has been designated as a free-running counter, and settings have been made so that high is output by compare match A and low is output by compare match B. When the pin level is the same as the specified level, the pin level does not change.

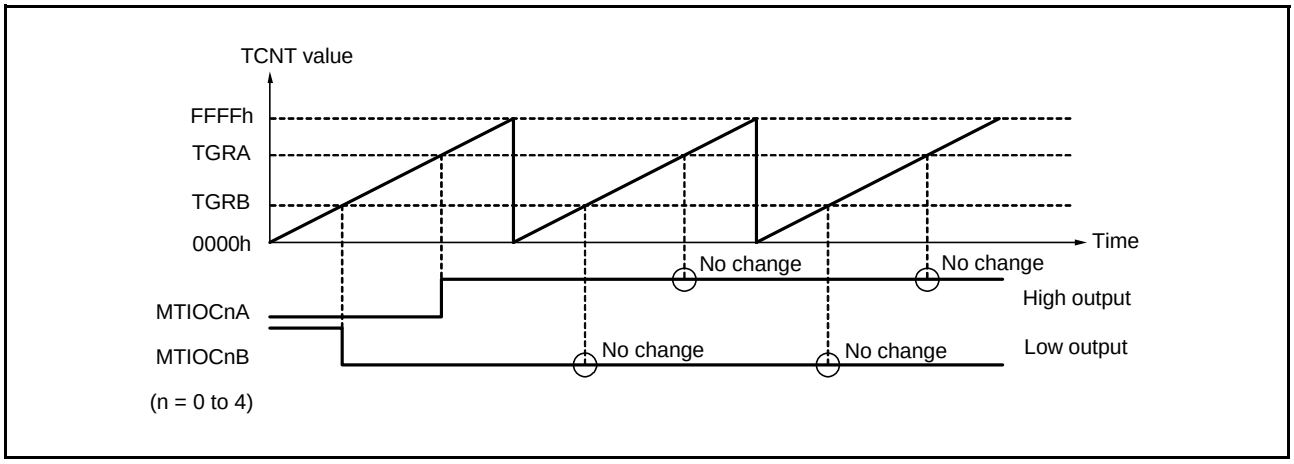



**Figure 20.8 Example of Low Output and High Output Operation**

Figure 20.9 shows an example of toggle output.
In this example, TCNT has been designated as a periodic counter (with counter clearing on compare match B), and settings have been made so that the output is toggled by both compare match A and compare match B.

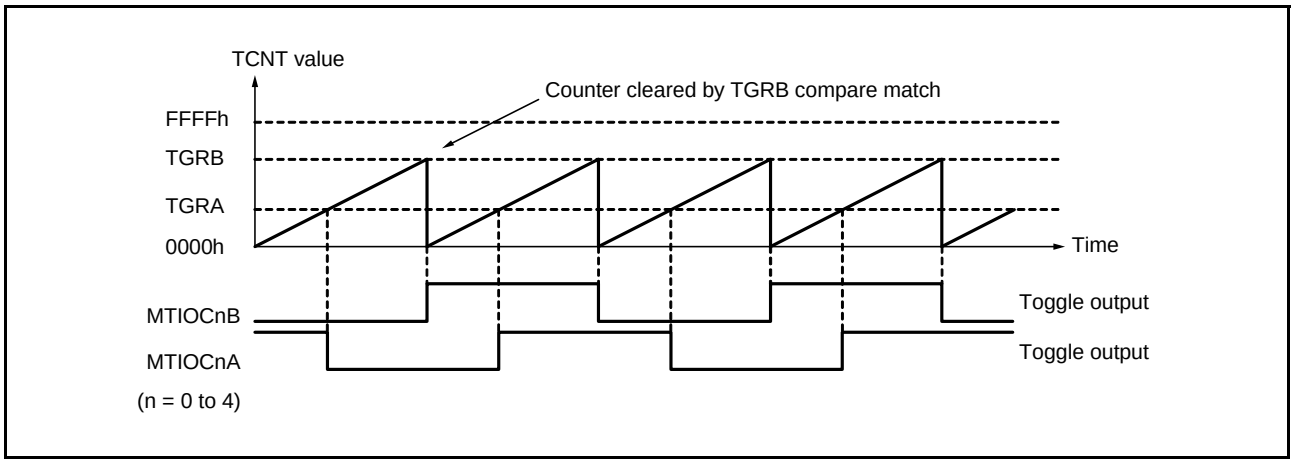



**Figure 20.9 Example of Toggle Output Operation**

### (3) Input Capture Function

The TCNT value can be transferred to TGR on detection of the input edge of the MTIOCnm (n = 0 to 4; m = A to D) pin and MTIC5m (m = W, V, U) pin.

The rising edge, falling edge, or both edges can be selected as the detection edge. For MTU0 and MTU1, another channel's counter input clock or compare match signal can also be specified as the input capture source.

Note: • When another channel's counter input clock is used as the input capture input for MTU0 and MTU1, PCLK/1 clock should not be selected as the counter input clock used for input capture input. Input capture will not be generated if PCLK/1 clock is selected.

#### (a) Example of Input Capture Operation Setting Procedure

Figure 20.10 shows an example of the input capture operation setting procedure.

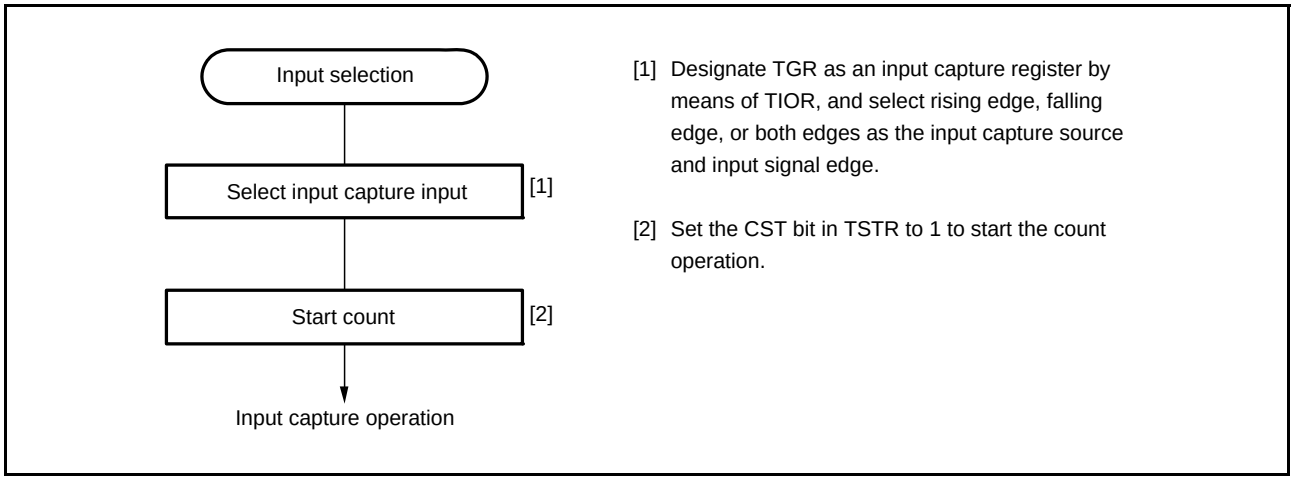



**Figure 20.10 Example of Input Capture Operation Setting Procedure**

### (b) Example of Input Capture Operation

Figure 20.11 shows an example of input capture operation.
In this example, both rising and falling edges have been selected as the MTIOCnA pin input capture input edge, the falling edge has been selected as the MTIOCnB pin input capture input edge, and counter clearing by TGRB input capture has been designated for TCNT.

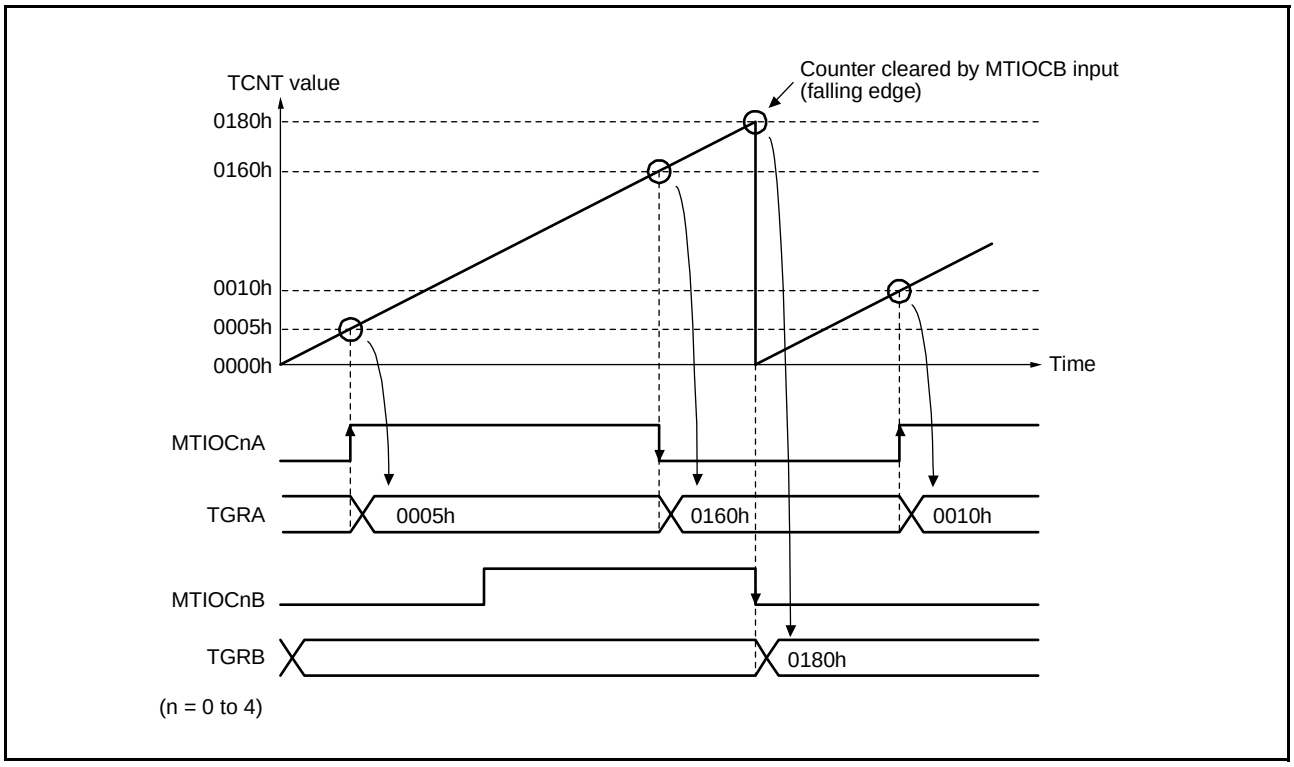


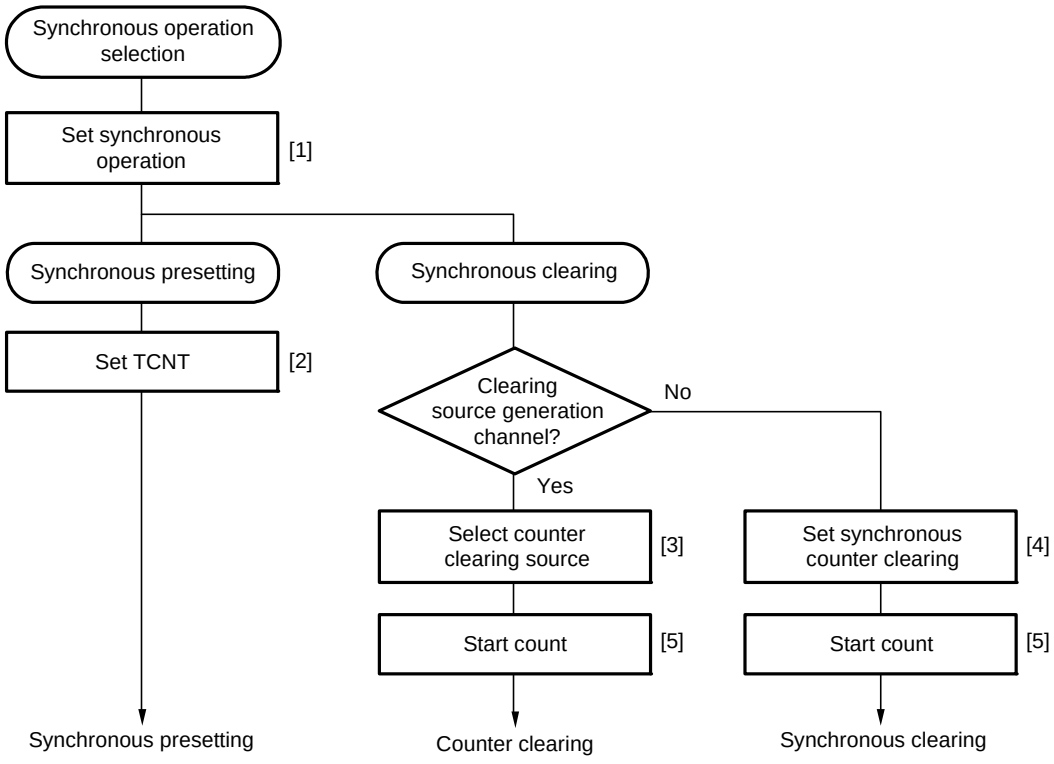

**Figure 20.11 Example of Input Capture Operation**

## 20.3.2 Synchronous Operation

In synchronous operation, the values in multiple TCNT counters can be modified simultaneously (synchronous presetting). In addition, multiple TCNT counters can be cleared simultaneously (synchronous clearing) by making the appropriate setting in TCR.
Synchronous operation increases the number of TGR registers assigned to a single time base.
MTU0 to MTU4 can all be designated for synchronous operation.
MTU5 cannot be used for synchronous operation.

### (1) Example of Synchronous Operation Setting Procedure

Figure 20.12 shows an example of the synchronous operation setting procedure.




[1] Set 1 in the SYNCn bit (n = 0 to 4) in TSYR corresponding to the channels to be designated for synchronous operation.

[2] When the TCNT counter of any of the channels designated for synchronous operation is written to, the same value is simultaneously written to the other TCNT counters.

[3] Use the CCLR[2:0] bits in TCR to specify TCNT clearing by input capture/output compare, etc.

[4] Use the CCLR[2:0] bits in TCR to designate synchronous clearing for the counter clearing source.

[5] Set 1 in the CST bit in TSTR for the relevant channels, to start the count operation.

**Figure 20.12 Example of Synchronous Operation Setting Procedure**

### (2) Example of Synchronous Operation

Figure 20.13 shows an example of synchronous operation.

In this example, synchronous operation and PWM mode 1 have been designated for MTU0 to MTU2, MTU0.TGRB compare match has been set as the counter clearing source in MTU0, and synchronous clearing has been set for the counter clearing source in MTU1 and MTU2.

Three-phase PWM waveforms are output from pins MTIOC0A, MTIOC1A, and MTIOC2A. At this time, synchronous presetting and synchronous clearing by MTU0.TGRB compare match are performed for the TCNT counters in MTU0 to MTU2, and the data set in MTU0.TGRB is used as the PWM cycle.

For details of PWM modes, refer to section 20.3.5, PWM Modes.

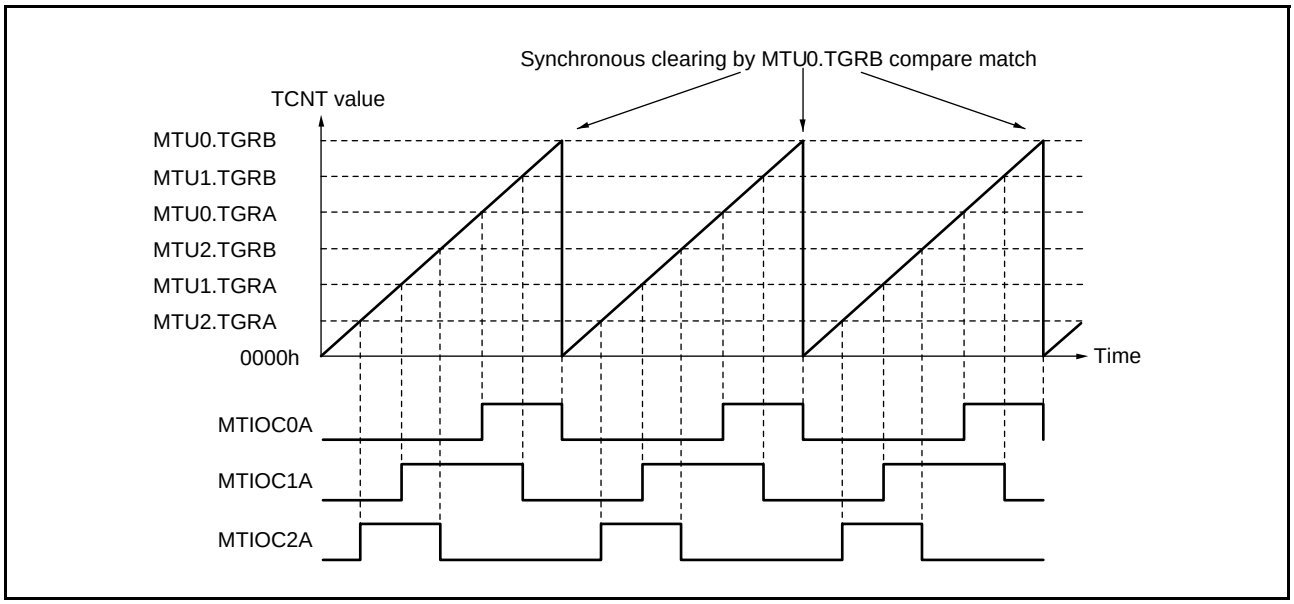



**Figure 20.13 Example of Synchronous Operation**

### 20.3.3 Buffer Operation

Buffer operation, provided for MTU0, MTU3, and MTU4, enables TGRC and TGRD to be used as buffer registers. In MTU0, TGRF can also be used as a buffer register.

Buffer operation differs depending on whether TGR has been designated as an input capture register or as a compare match register.

Note: • MTU0.TGRE cannot be designated as an input capture register and can only operate as a compare match register.

Table 20.43 shows the register combinations used in buffer operation.

**Table 20.43 Register Combinations in Buffer Operation**

| Channel | Timer General Register | Buffer Register |
|---|---|---|
| MTU0 | TGRA | TGRC |
| | TGRB | TGRD |
| | TGRE | TGRF |
| MTU3 | TGRA | TGRC |
| | TGRB | TGRD |
| MTU4 | TGRA | TGRC |
| | TGRB | TGRD |

- When TGR is an output compare register

When a compare match occurs, the value in the buffer register for the corresponding channel is transferred to the timer general register.

This operation is illustrated in Figure 20.14.

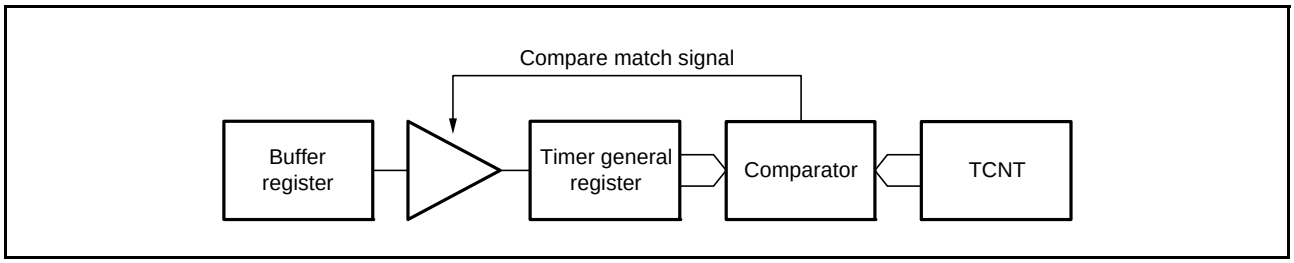



**Figure 20.14 Compare Match Buffer Operation**

- When TGR is an input capture register

When an input capture occurs, the value in TCNT is transferred to TGR and the value previously held in TGR is transferred to the buffer register.
This operation is illustrated in Figure 20.15.

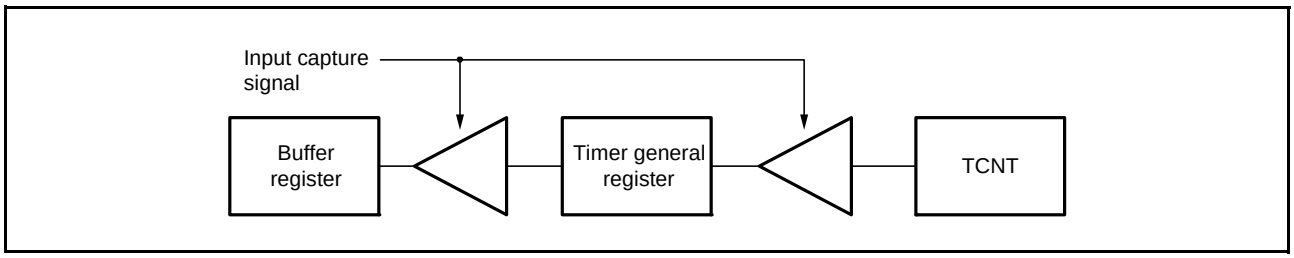



**Figure 20.15 Input Capture Buffer Operation**

## (1) Example of Buffer Operation Setting Procedure

Figure 20.16 shows an example of the buffer operation setting procedure.

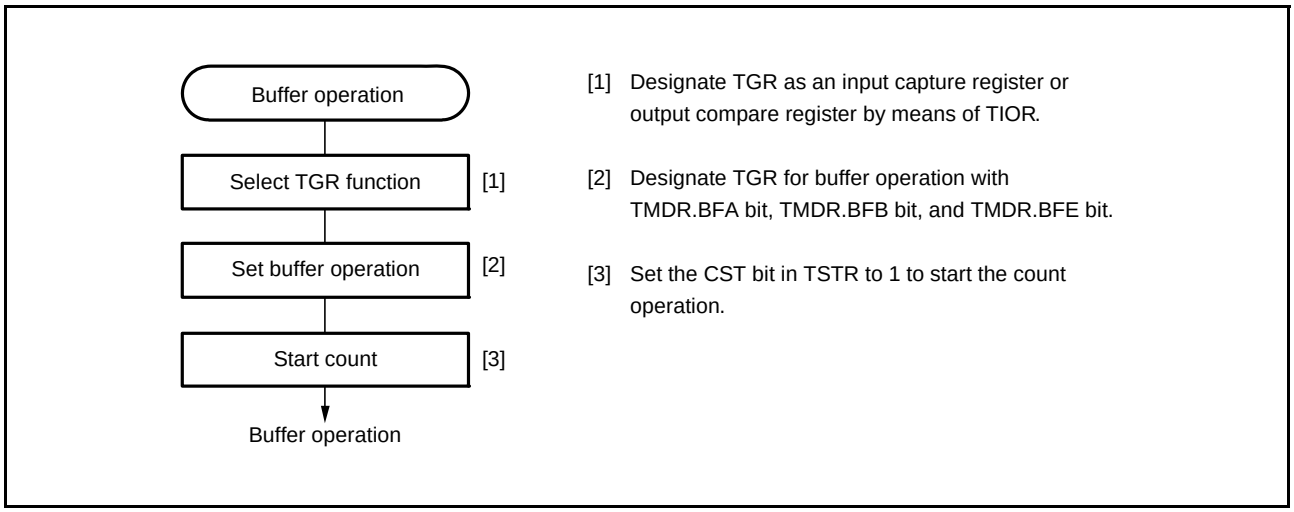



**Figure 20.16 Example of Buffer Operation Setting Procedure**

### (2) Examples of Buffer Operation

#### (a) When TGR is an Output Compare Register

Figure 20.17 shows an operation example in which PWM mode 1 has been designated for MTU0, and buffer operation has been designated for TGRA and TGRC. The settings used in this example are TCNT clearing by compare match B, high output at compare match A, and low output at compare match B. In this example, the TTSA bit in TBTM is cleared to 0.

As buffer operation has been set, when compare match A occurs, the output changes and the value in buffer register TGRC is simultaneously transferred to timer general register TGRA. This operation is repeated each time compare match A occurs.

For details of PWM modes, refer to section 20.3.5, PWM Modes.

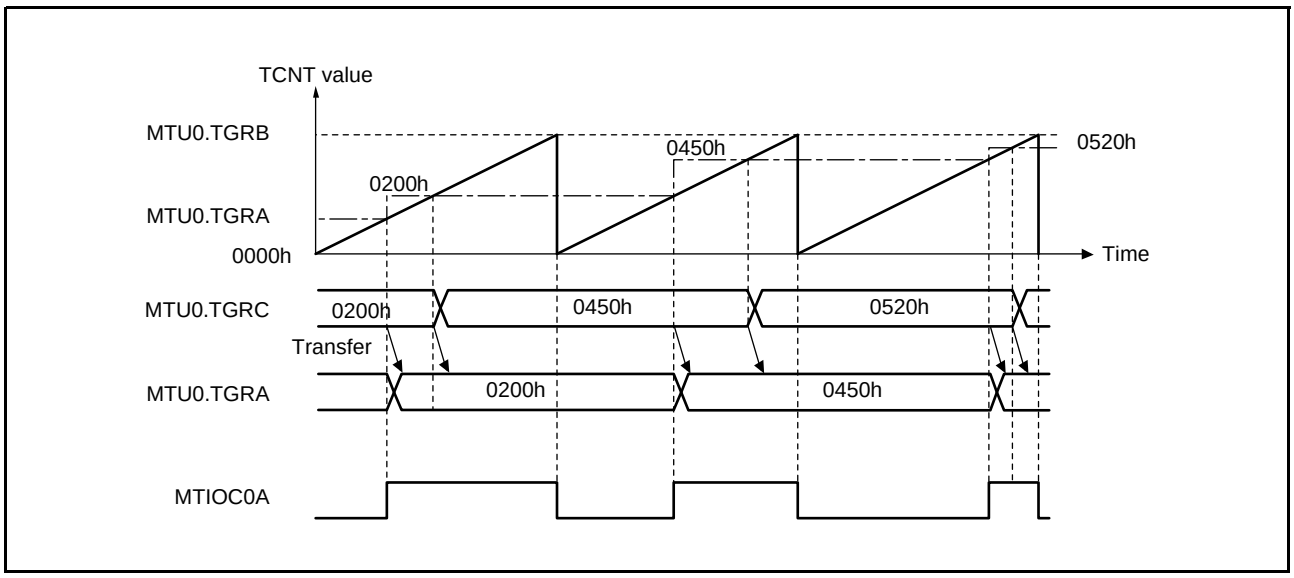



**Figure 20.17 Example of Buffer Operation (1)**

### (b) When TGR is an Input Capture Register

Figure 20.18 shows an operation example in which TGRA has been designated as an input capture register, and buffer operation has been designated for TGRA and TGRC.
Counter clearing by TGRA input capture has been set for TCNT, and both rising and falling edges have been selected as the MTIOCnA pin input capture input edge.
As buffer operation has been set, when the TCNT value is stored in TGRA upon occurrence of input capture A, the value previously stored in TGRA is simultaneously transferred to TGRC.

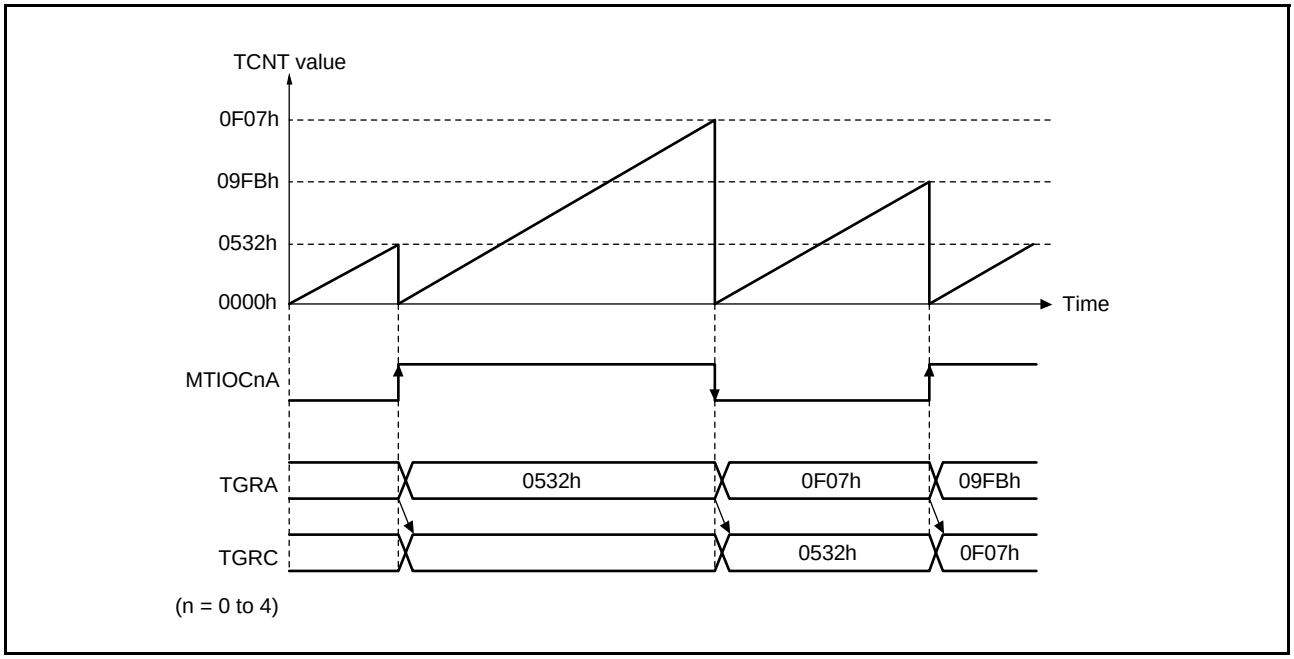



**Figure 20.18 Example of Buffer Operation (2)**

### (3) Selecting Timing for Transfer from Buffer Registers to Timer General Registers in Buffer Operation

The timing for transfer from buffer registers to timer general registers can be selected in PWM mode 1 or 2 for MTU0 or in PWM mode 1 for MTU3 and MTU4 by setting the timer buffer operation transfer mode registers (MTU0.TBTM, MTU3.TBTM, and MTU4.TBTM). Either compare match (initial setting) or TCNT clearing can be selected for the transfer timing. TCNT clearing as transfer timing is one of the following cases.

- When TCNT overflows (FFFFh → 0000h)
- When 0000h is written to TCNT during counting
- When TCNT is cleared to 0000h under the condition specified in the CCLR[2:0] bits in TCR

Note: • TBTM must be modified only while TCNT stops.

Figure 20.19 shows an operation example in which PWM mode 1 is designated for MTU0 and buffer operation is designated for MTU0.TGRA and MTU0.TGRC. The settings used in this example are MTU0.TCNT clearing by compare match B, high output at compare match A, and low output at compare match B. The TTSA bit in MTU0.TBTM is set to 1.

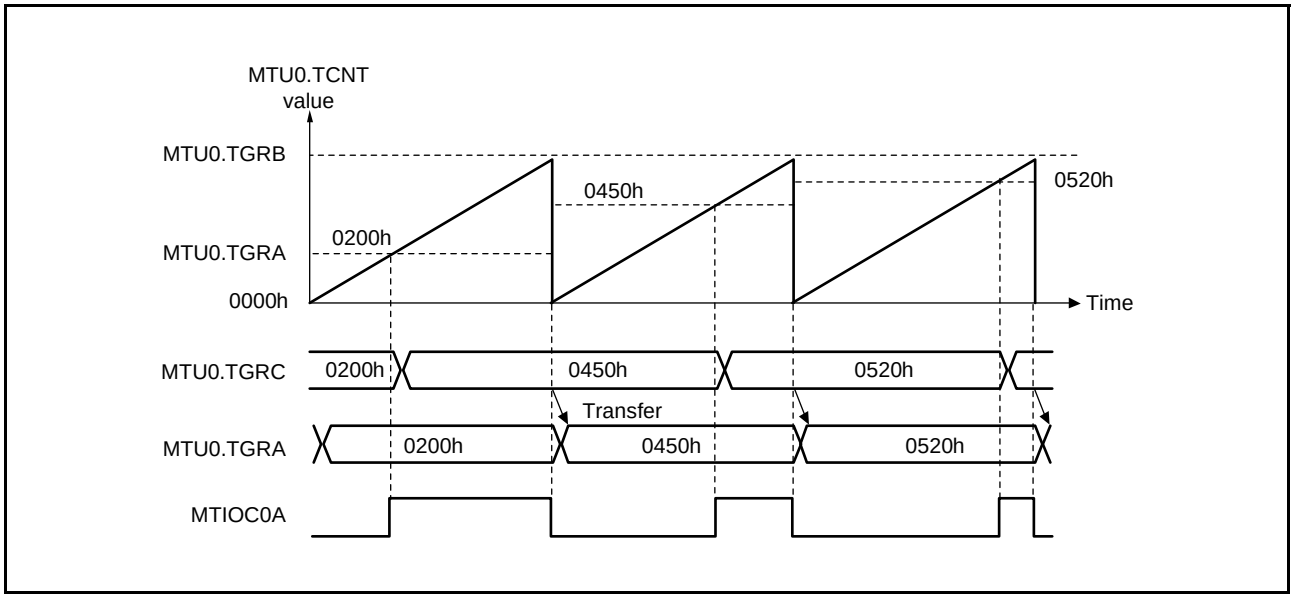



**Figure 20.19 Example of Buffer Operation When MTU0.TCNT Clearing is Selected for MTU0.TGRC-to-MTU0.TGRA Transfer Timing**

### 20.3.4 Cascaded Operation

In cascaded operation, 16-bit counters in different two channels are used together as a 32-bit counter.
This function works when overflow/underflow of MTU2.TCNT is selected as the counter clock for MTU1 through the TPSC[2:0] bits in TCR.
Underflow occurs only when the lower 16 bits of TCNT is in phase counting mode.
Table 20.44 lists the register combinations used in cascaded operation.

Note: • When phase counting mode is set for MTU1 or MTU2, the counter clock setting is invalid and the counters operate independently in phase counting mode.

**Table 20.44 Cascaded Combinations**

| Combination | Upper 16 Bits | Lower 16 Bits |
|---|---|---|
| MTU1 and MTU2 | MTU1.TCNT | MTU2.TCNT |

For simultaneous input capture of MTU1.TCNT and MTU2.TCNT during cascaded operation, additional input capture input pins can be specified by the timer input capture control register (TICCR). The input-capture condition is of edges in the signal produced by taking the logical OR of the input level on the main input pin and the input level on the added input pin. Accordingly, if either is at the high, a change in the level of the other will not produce an edge for detection. For details, see (4) Cascaded Operation Example (c). For input capture in cascade connection, refer to section 20.6.22, Simultaneous Input Capture in MTU1.TCNT and MTU2.TCNT in Cascade Connection.
Table 20.45 lists the TICCR setting and input capture input pins.

**Table 20.45 TICCR Setting and Input Capture Input Pins**

| Target Input Capture | TICCR Setting | Input Capture Input Pin |
|---|---|---|
| Input capture from MTU1.TCNT to MTU1.TGRA | I2AE bit = 0 (initial value) | MTIOC1A |
| | I2AE bit = 1 | MTIOC1A, MTIOC2A |
| Input capture from MTU1.TCNT to MTU1.TGRB | I2BE bit = 0 (initial value) | MTIOC1B |
| | I2BE bit = 1 | MTIOC1B, MTIOC2B |
| Input capture from MTU2.TCNT to MTU2.TGRA | I1AE bit = 0 (initial value) | MTIOC2A |
| | I1AE bit = 1 | MTIOC2A, MTIOC1A |
| Input capture from MTU2.TCNT to MTU2.TGRB | I1BE bit = 0 (initial value) | MTIOC2B |
| | I1BE bit = 1 | MTIOC2B, MTIOC1B |

### (1) Example of Cascaded Operation Setting Procedure

Figure 20.20 shows an example of the cascaded operation setting procedure.

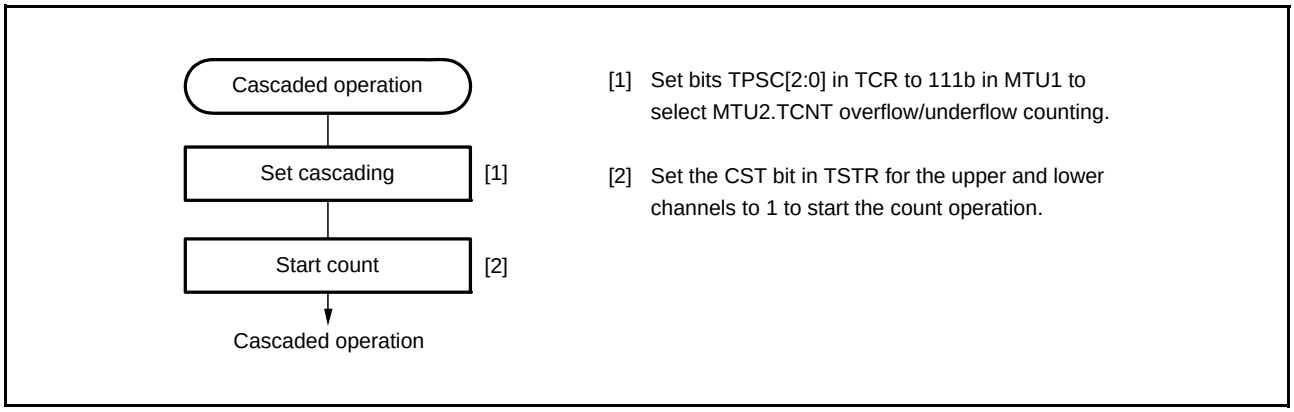



**Figure 20.20 Cascaded Operation Setting Procedure**

### (2) Cascaded Operation Example (a)

Figure 20.21 illustrates the operation when MTU2.TCNT overflow/underflow counting has been set for MTU1.TCNT and phase counting mode has been designated for MTU2.
MTU1.TCNT is incremented by MTU2.TCNT overflow and decremented by MTU2.TCNT underflow.

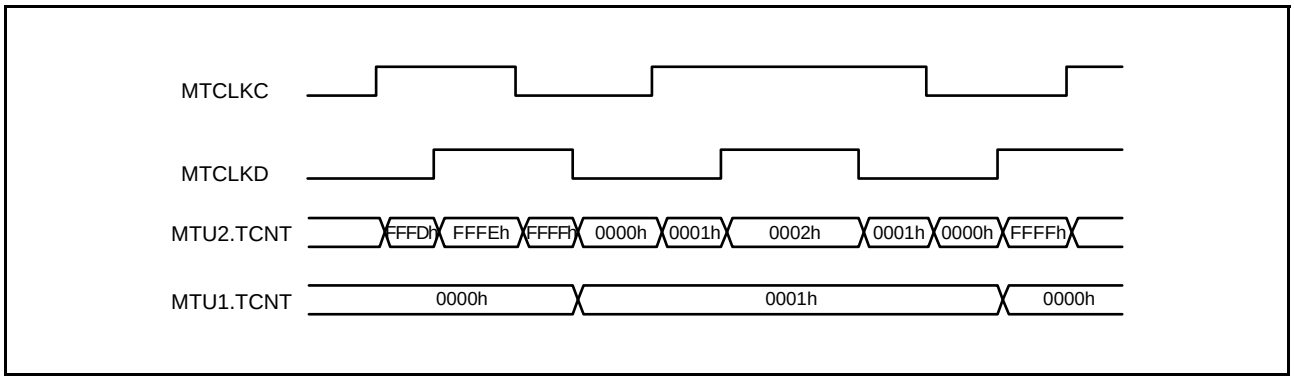



**Figure 20.21 Cascaded Operation Example (a)**

### (3) Cascaded Operation Example (b)

Figure 20.22 illustrates the operation when MTU1.TCNT and MTU2.TCNT have been cascaded and the TICCR.I2AE bit has been set to 1 to include the MTIOC2A pin in the MTU1.TGRA input capture conditions. In this example, the IOA[3:0] bits in MTU1.TIOR have selected the MTIOC1A rising edge for the input capture timing while the IOA[3:0] bits in MTU2.TIOR have selected the MTIOC2A rising edge for the input capture timing.
Under these conditions, the rising edge of both MTIOC1A and MTIOC2A is used for the MTU1.TGRA input capture condition. For the MTU2.TGRA input capture condition, the MTIOC2A rising edge is used.

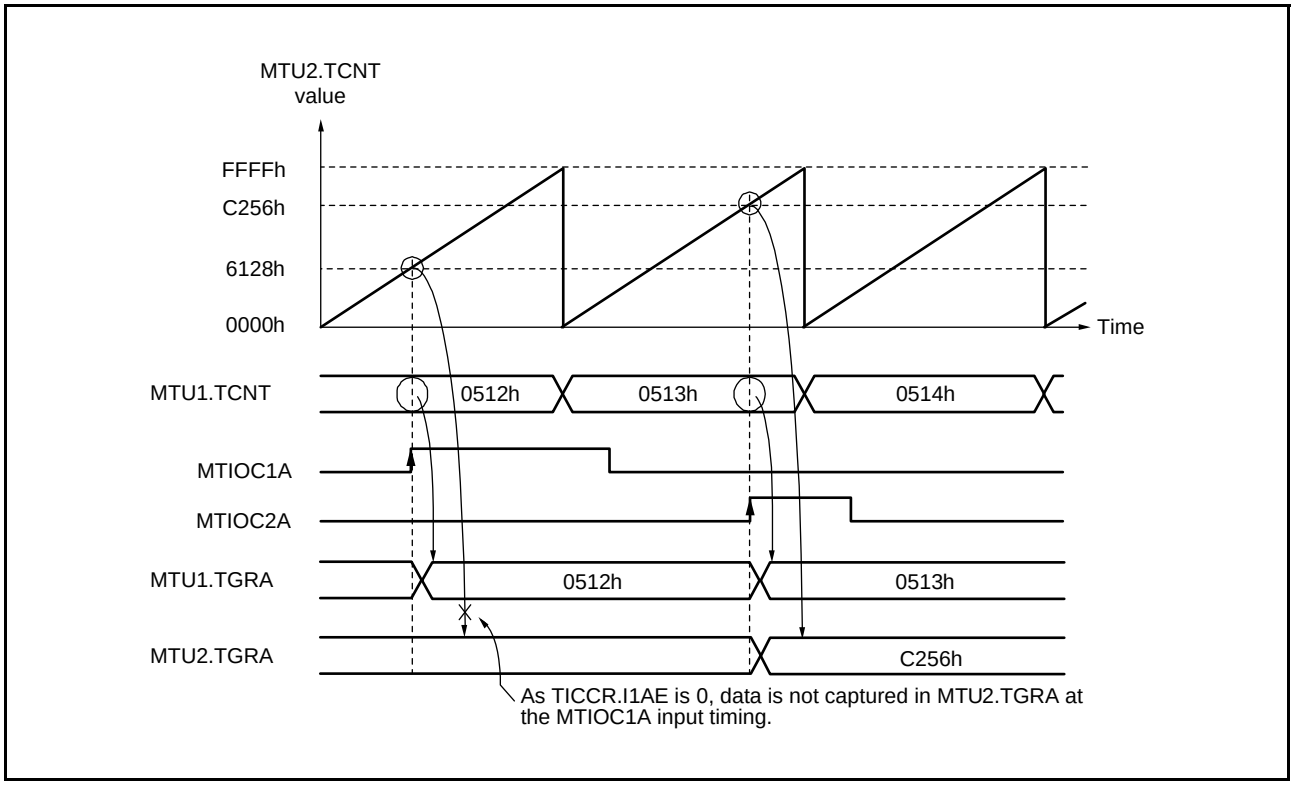
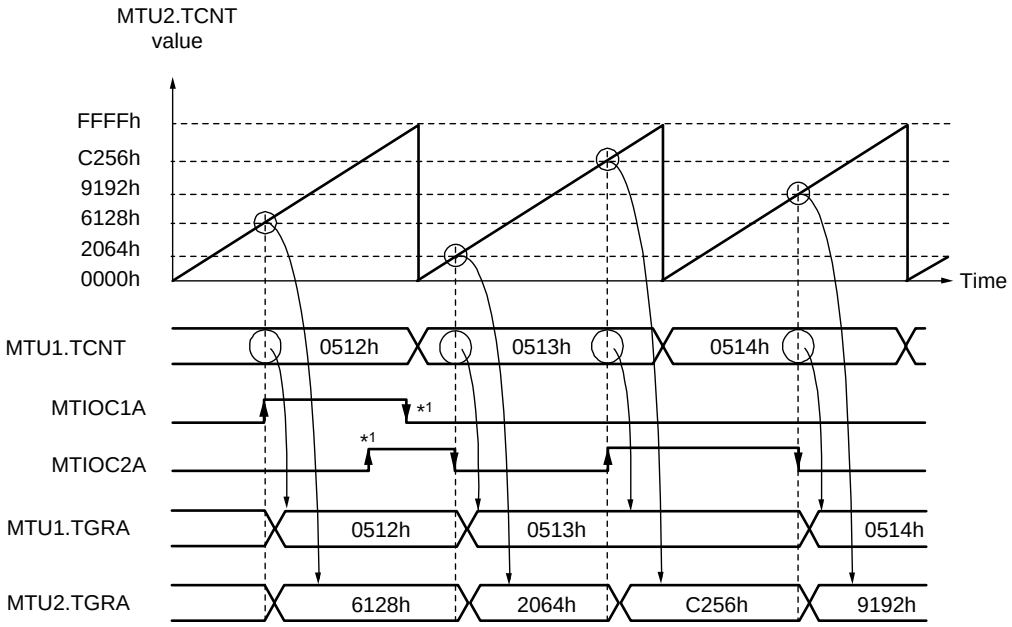



**Figure 20.22 Cascaded Operation Example (b)**

### (4) Cascaded Operation Example (c)

Figure 20.23 illustrates the operation when MTU1.TCNT and MTU2.TCNT have been cascaded and the I2AE and I1AE bits in TICCR have been set to 1 to include the MTIOC2A and MTIOC1A pins in the MTU1.TGRA and MTU2.TGRA input capture conditions, respectively. In this example, the IOA[3:0] bits in both MTU1.TIOR and MTU2.TIOR have selected both the rising and falling edges for the input capture timing. Under these conditions, the OR result of MTIOC1A and MTIOC2A input is used for the MTU1.TGRA and MTU2.TGRA input capture conditions.




Note 1. When either input signal is at the high level, an edge in the other input signal does not act as the condition for capture.

**Figure 20.23 Cascaded Operation Example (c)**

### (5) Cascaded Operation Example (d)

Figure 20.24 illustrates the operation when MTU1.TCNT and MTU2.TCNT have been cascaded and the I2AE bit in TICCR has been set to 1 to include the MTIOC2A pin in the MTU1.TGRA input capture conditions. In this example, the IOA[3:0] bits in MTU1.TIOR have selected occurrence of MTU0.TGRA compare match or input capture for the input capture timing while the IOA[3:0] bits in MTU2.TIOR have selected the MTIOC2A rising edge for the input capture timing.

Under these conditions, as MTU1.TIOR has selected occurrence of MTU0.TGRA compare match or input capture for the input capture timing, the MTIOC2A edge is not used for MTU1.TGRA input capture condition although the I2AE bit in TICCR has been set to 1.

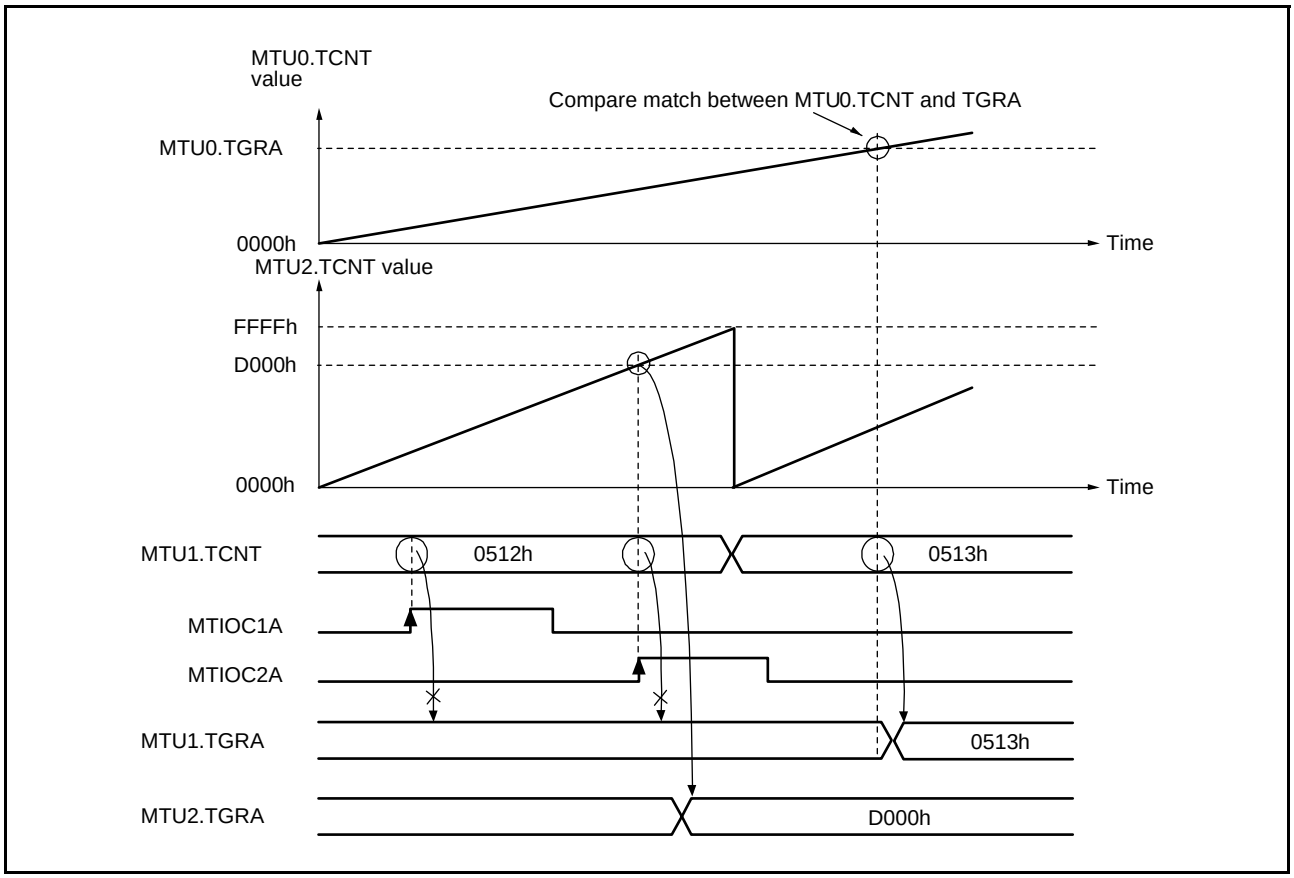



**Figure 20.24 Cascaded Operation Example (d)**

### 20.3.5 PWM Modes

PWM modes are provided to output PWM waveforms from the external pins. The output level can be selected as low, high, or toggle output in response to a compare match of each TGR.
PWM waveforms in the range of 0% to 100% duty cycle can be output according to the TGR settings.
By designating TGR compare match as the counter clearing source, the PWM cycle can be specified in that register.
Each channel can be set to PWM mode independently. Synchronous operation is also possible.
There are two PWM modes as described below.

#### (a) PWM Mode 1

PWM waveforms are output from the MTIOCnA and MTIOCnC pins by pairing TGRA with TGRB and TGRC with TGRD. The levels specified by bits IOA[3:0] and IOC[3:0] in TIOR are output from the MTIOCnA and MTIOCnC pins at compare matches A and C, and the levels specified by bits IOB[3:0] and IOD[3:0] in TIOR are output at compare matches B and D. The initial output value is set in TGRA or TGRC. If the values set in paired TGRs are identical, the output value does not change even when a compare match occurs.
In PWM mode 1, up to eight phases of PWM waveforms can be output.

#### (b) PWM Mode 2

PWM output is generated using one TGR as the cycle register and the others as duty registers. The level specified in TIOR is output at compare matches. Upon counter clearing by a cycle register compare match, the initial value set in TIOR is output from each pin. If the values set in the cycle and duty registers are identical, the output value does not change even when a compare match occurs.
In PWM mode 2, up to eight phases of PWM waveforms can be output when using synchronous operation in combination.
The correspondence between PWM output pins and registers is listed in Table 20.46.

**Table 20.46 PWM Output Registers and Output Pins**

| Channel | Register | Output Pins | |
|---|---|---|---|
| | | PWM Mode 1 | PWM Mode 2 |
| MTU0 | MTU0.TGRA | MTIOC0A | MTIOC0A |
| | MTU0.TGRB | | MTIOC0B |
| | MTU0.TGRC | MTIOC0C | MTIOC0C |
| | MTU0.TGRD | | MTIOC0D |
| MTU1 | MTU1.TGRA | MTIOC1A | MTIOC1A |
| | MTU1.TGRB | | MTIOC1B |
| MTU2 | MTU2.TGRA | MTIOC2A | MTIOC2A |
| | MTU2.TGRB | | MTIOC2B |
| MTU3 | MTU3.TGRA | MTIOC3A | Setting prohibited |
| | MTU3.TGRB | | |
| | MTU3.TGRC | MTIOC3C | |
| | MTU3.TGRD | | |
| MTU4 | MTU4.TGRA | MTIOC4A | |
| | MTU4.TGRB | | |
| | MTU4.TGRC | MTIOC4C | |
| | MTU4.TGRD | | |

Note: • In PWM mode 2, PWM output is not possible for the TGR register in which the PWM cycle is set.

### (1) Example of PWM Mode Setting Procedure

Figure 20.25 shows an example of the PWM mode setting procedure.

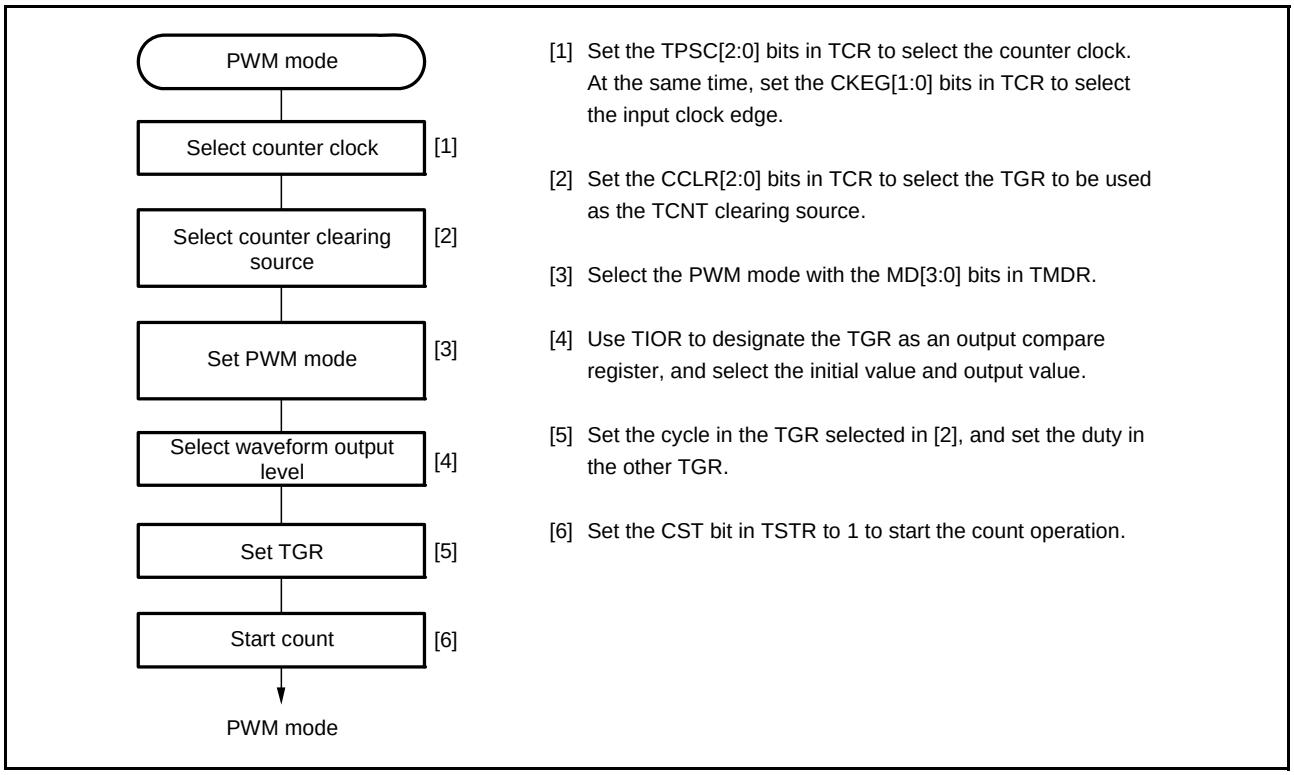



**Figure 20.25 Example of PWM Mode Setting Procedure**

### (2) Examples of PWM Mode Operation

Figure 20.26 shows an example of operation in PWM mode 1.
In this example, TGRA compare match is set as the TCNT clearing source, a low level is set as the initial output value and output value for TGRA, and a high level is set as the output value for TGRB.
In this case, the value set in TGRA is used as the cycle, and the value set in TGRB is used as the duty.

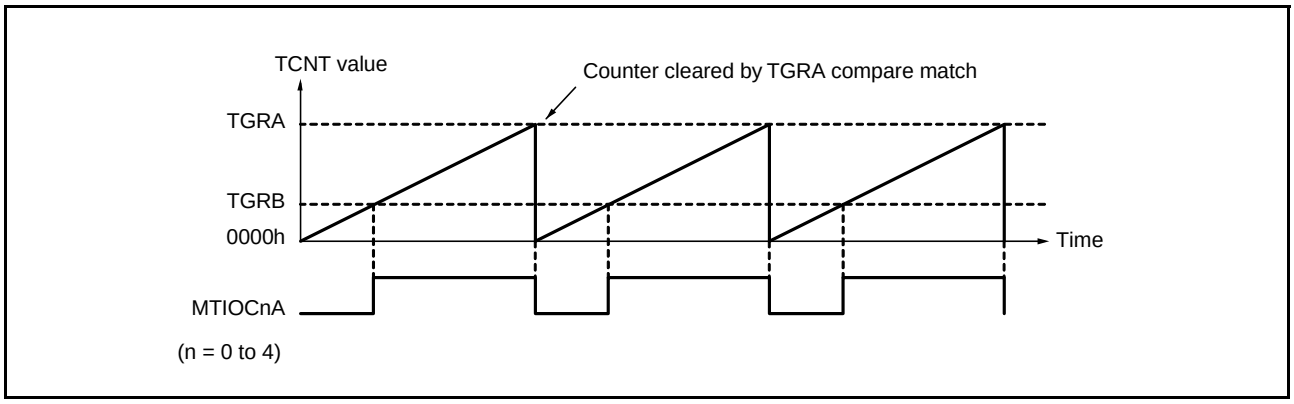



**Figure 20.26 Example of PWM Mode Operation**

Figure 20.27 shows an example of operation in PWM mode 2.
In this example, synchronous operation is designated for MTU0 and MTU1, MTU1.TGRB compare match is set as the TCNT clearing source, and a low level is set as the initial output value and a high level as the output value for the other TGR registers (MTU0.TGRA to MTU0.TGRD and MTU1.TGRA), outputting 5-phase PWM waveforms.
In this case, the value set in MTU1.TGRB is used as the cycle, and the values set in the other TGRs are used as the duty.

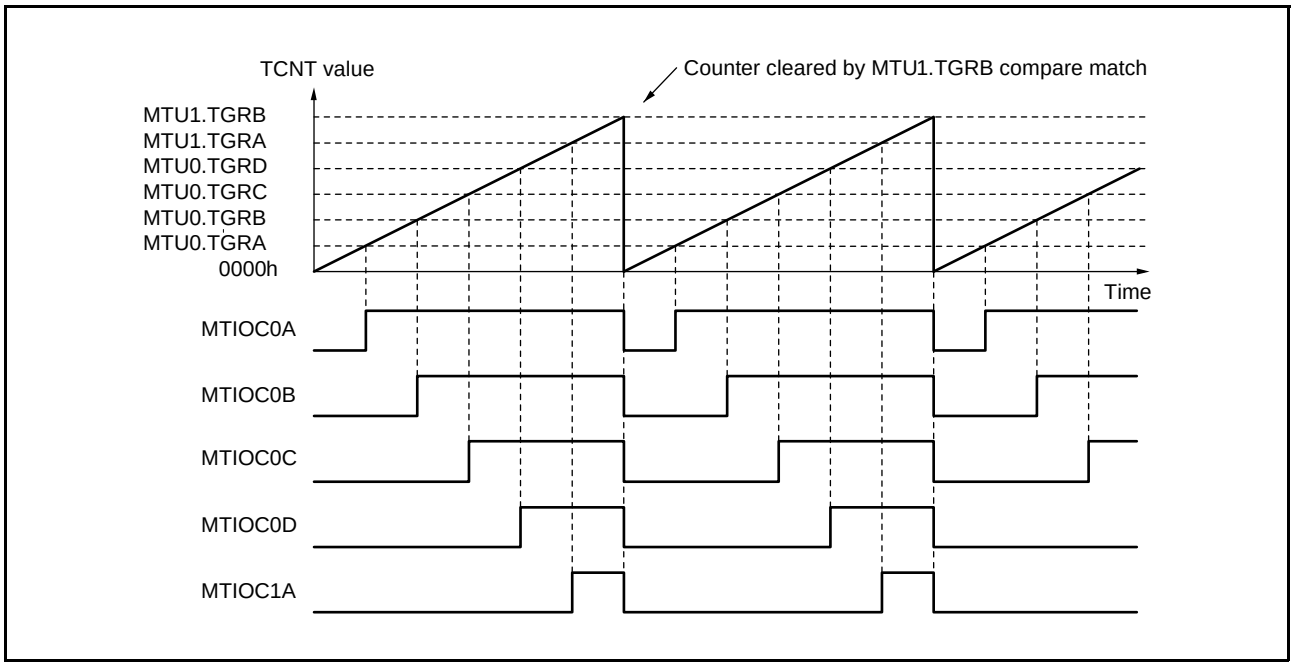



**Figure 20.27 Example of PWM Mode Operation**

Figure 20.28 shows examples of PWM waveform output with 0% duty and 100% duty in PWM mode.

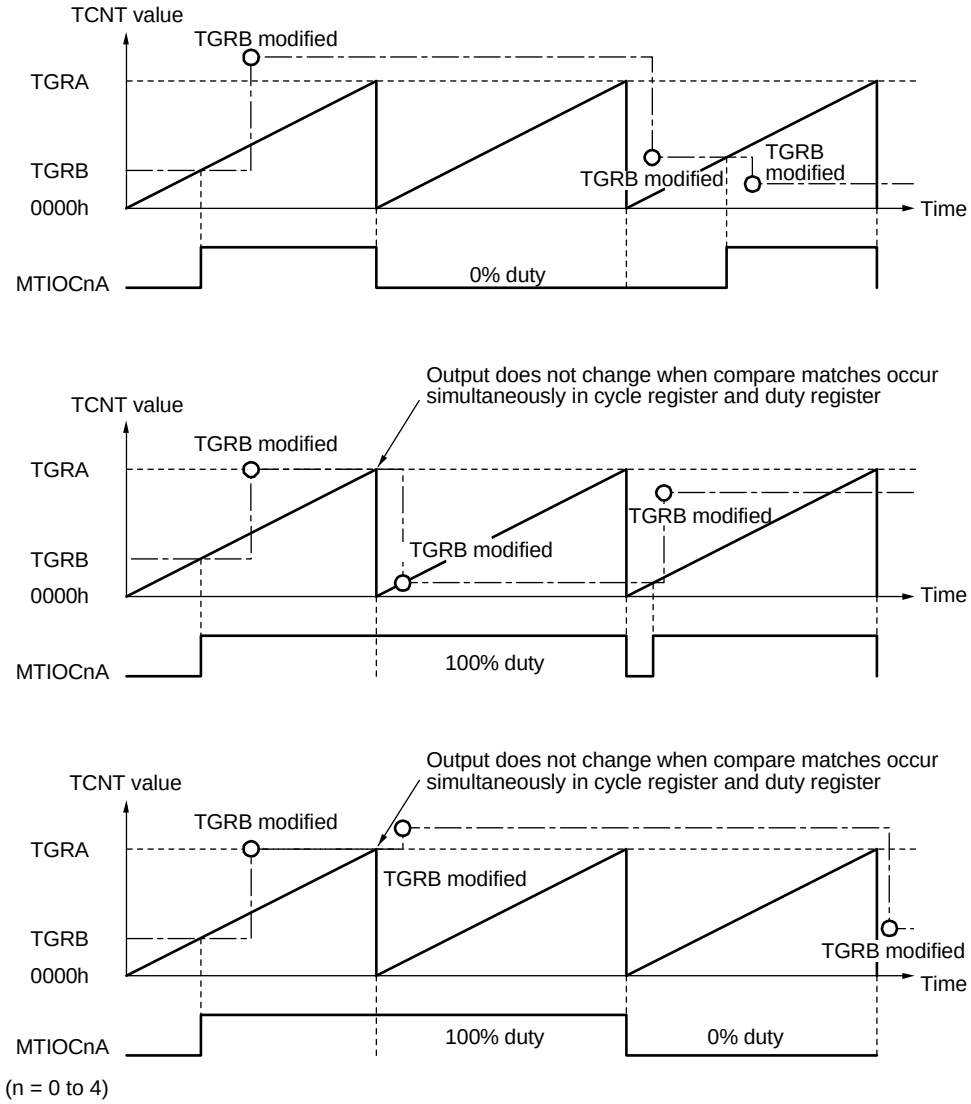



**Figure 20.28 Examples of PWM Mode Operation**

### 20.3.6 Phase Counting Mode

When phase counting mode is specified, an external clock is selected as the counter input clock and TCNT operates as an up-counter/down-counter regardless of the setting of the TCR.TPSC[2:0] bits and TCR.CKEG[1:0] bits. However, the functions of the TCR.CCLR[2:0] bits and of TIOR, TIER, and TGR are valid, and input capture/compare match and interrupt functions can be used.
This can be used for two-phase encoder pulse input.
If an overflow occurs while TCNT is counting up, a TCIV interrupt is generated while the corresponding TIER.TCIEV bit is 1. If an underflow occurs while TCNT is counting down, a TCIU interrupt is generated while the corresponding TIER.TCIEU bit is 1.
The TSR.TCFD flag is the count direction flag. Read the TCFD flag to check whether TCNT is counting up or down.
Table 20.47 lists the correspondence between external clock pins and channels.

**Table 20.47 Clock Input Pins in Phase Counting Mode**

| | External Clock Input Pins | |
|---|---|---|
| **Channel** | **A-Phase** | **B-Phase** |
| MTU1 | MTCLKA | MTCLKB |
| MTU2 | MTCLKC | MTCLKD |

#### (1) Example of Phase Counting Mode Setting Procedure

Figure 20.29 shows an example of the phase counting mode setting procedure.

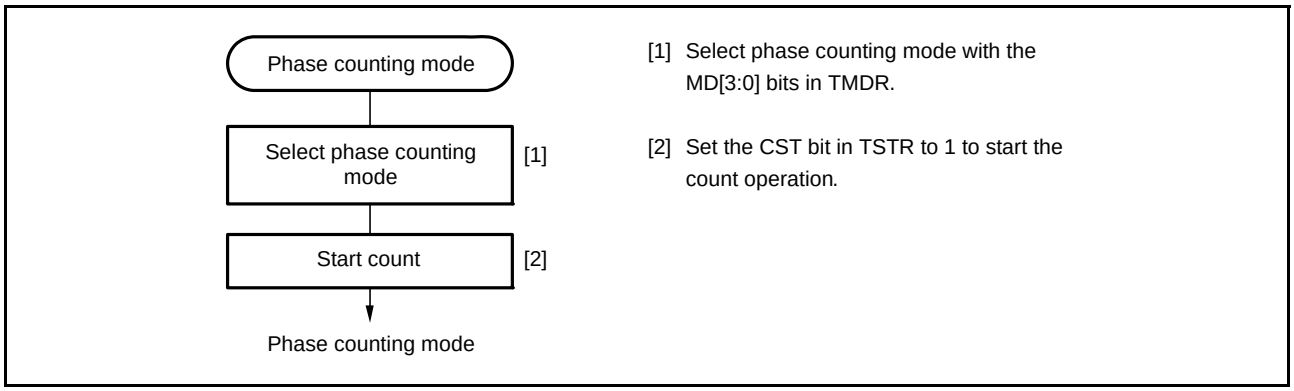



**Figure 20.29 Example of Phase Counting Mode Setting Procedure**

### (2) Examples of Phase Counting Mode Operation

In phase counting mode, TCNT is incremented or decremented according to the phase difference between two external clocks. There are four modes according to the count conditions.

#### (a) Phase Counting Mode 1

Figure 20.30 shows an example of operation in phase counting mode 1, and Table 20.48 lists the TCNT up-count/down-count conditions.

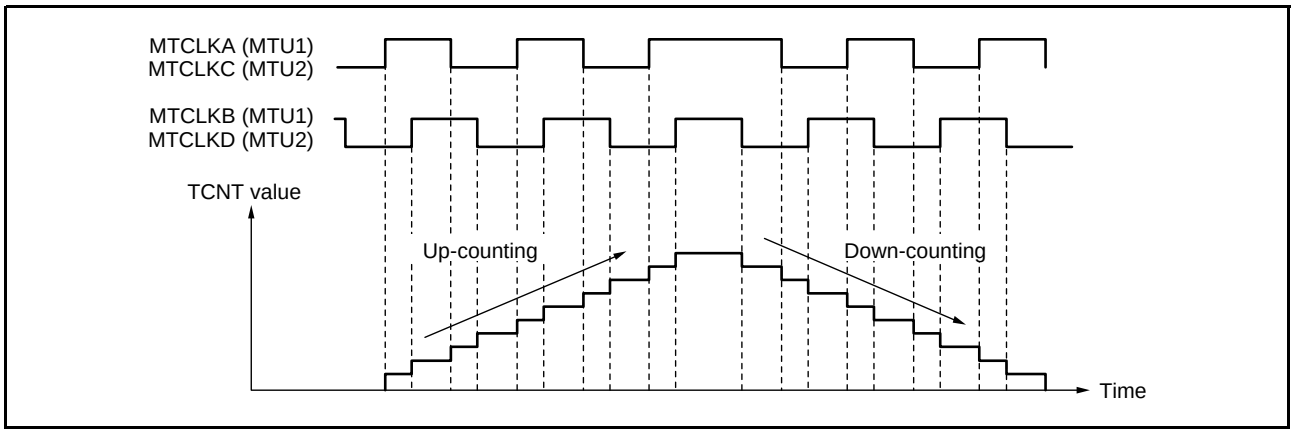



**Figure 20.30 Example of Operation in Phase Counting Mode 1**

**Table 20.48 Up-Count/Down-Count Conditions in Phase Counting Mode 1**

| MTCLKA (MTU1) MTCLKC (MTU2) | MTCLKB (MTU1) MTCLKD (MTU2) | Operation |
|---|---|---|
| High | ↑ | Up-counting |
| Low | ↓ | |
| ↑ | Low | |
| ↓ | High | |
| High | ↓ | Down-counting |
| Low | ↑ | |
| ↑ | High | |
| ↓ | Low | |

↑ : Rising edge

↓ : Falling edge

### (b) Phase Counting Mode 2

Figure 20.31 shows an example of operation in phase counting mode 2, and Table 20.49 lists the TCNT up-count/down-count conditions.

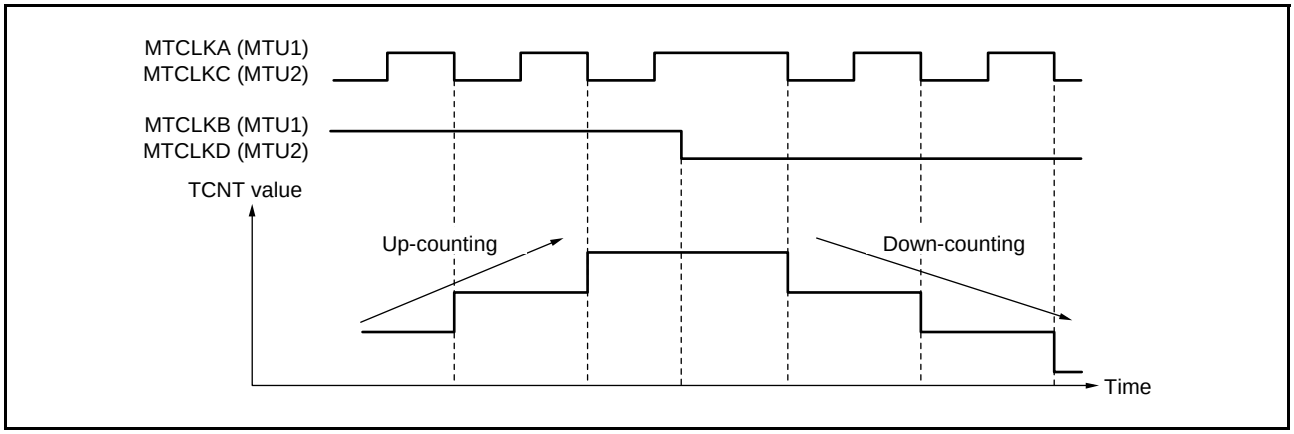



**Figure 20.31 Example of Operation in Phase Counting Mode 2**

**Table 20.49 Up-Count/Down-Count Conditions in Phase Counting Mode 2**

| MTCLKA (MTU1) MTCLKC (MTU2) | MTCLKB (MTU1) MTCLKD (MTU2) | Operation |
|---|---|---|
| High | ↑ | None (Don't care) |
| Low | ↓ | None (Don't care) |
| ↑ | Low | None (Don't care) |
| ↓ | High | Up-counting |
| High | ↓ | None (Don't care) |
| Low | ↑ | None (Don't care) |
| ↑ | High | None (Don't care) |
| ↓ | Low | Down-counting |

↑ : Rising edge

↓ : Falling edge

### (c) Phase Counting Mode 3

Figure 20.32 shows an example of operation in phase counting mode 3, and Table 20.50 lists the TCNT up-count/down-count conditions.

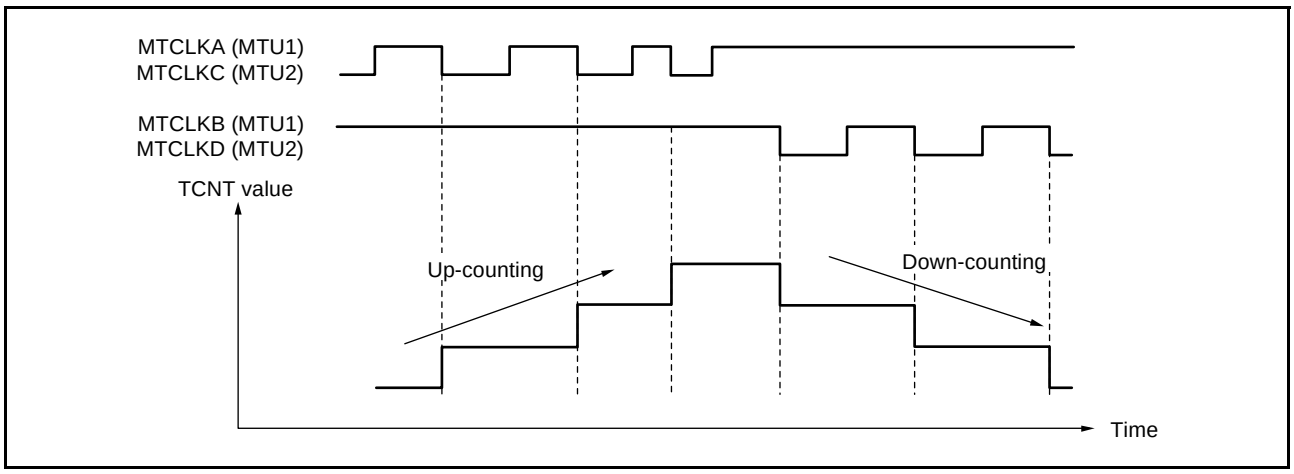



**Figure 20.32 Example of Operation in Phase Counting Mode 3**

**Table 20.50 Up-Count/Down-Count Conditions in Phase Counting Mode 3**

| MTCLKA (MTU1) MTCLKC (MTU2) | MTCLKB (MTU1) MTCLKD (MTU2) | Operation |
|---|---|---|
| High | ↑ | None (Don't care) |
| Low | ↓ | None (Don't care) |
| ↑ | Low | None (Don't care) |
| ↓ | High | Up-counting |
| High | ↓ | Down-counting |
| Low | ↑ | None (Don't care) |
| ↑ | High | None (Don't care) |
| ↓ | Low | None (Don't care) |

↑ : Rising edge

↓ : Falling edge

### (d) Phase Counting Mode 4

Figure 20.33 shows an example of operation in phase counting mode 4, and Table 20.51 lists the TCNT up-count/down-count conditions.

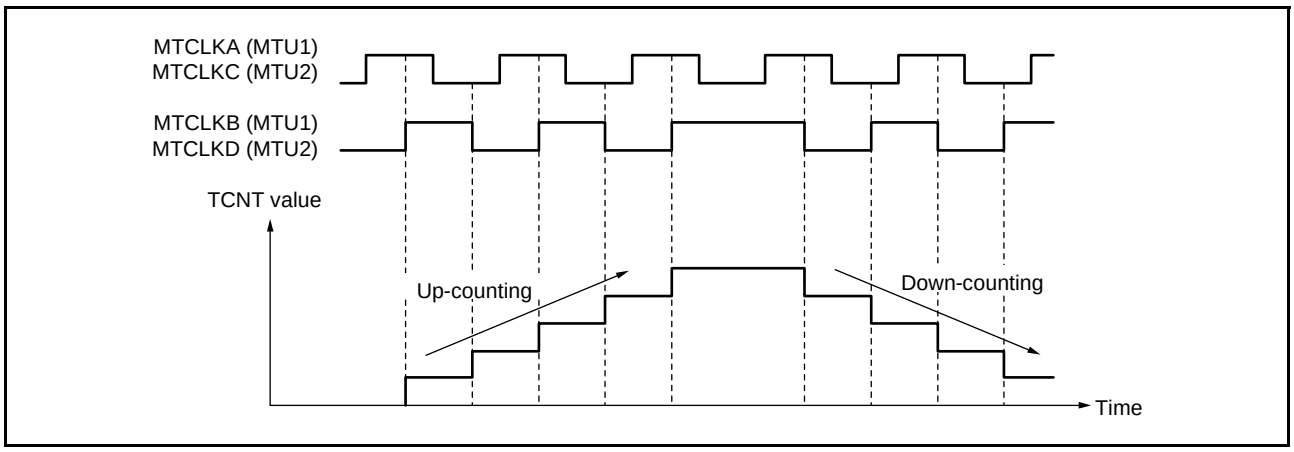



**Figure 20.33 Example of Operation in Phase Counting Mode 4**

**Table 20.51 Up-Count/Down-Count Conditions in Phase Counting Mode 4**

| MTCLKA (MTU1) MTCLKC (MTU2) | MTCLKB (MTU1) MTCLKD (MTU2) | Operation |
|---|---|---|
| High | ↑ | Up-counting |
| Low | ↓ | |
| ↑ | Low | None (Don't care) |
| ↓ | High | |
| High | ↓ | Down-counting |
| Low | ↑ | |
| ↑ | High | None (Don't care) |
| ↓ | Low | |

↑ : Rising edge

↓ : Falling edge

### (3) Phase Counting Mode Application Example

Figure 20.34 shows an example in which MTU1 is in phase counting mode, and MTU1 is coupled with MTU0 to input 2-phase encoder pulses of a servo motor in order to detect position or speed.

MTU1 is set to phase counting mode 1, and the encoder pulse A-phase and B-phase are input to MTCLKA and MTCLKB.

MTU0.TGRC compare match is specified as the MTU0.TCNT clearing source and MTU0.TGRA and TGRC are used for the compare match function and are set with the speed control cycle and position control cycle. MTU0.TGRB is used for input capture, with MTU0.TGRB and MTU0.TGRD operating in buffer mode. The MTU1 counter input clock is designated as the MTU0.TGRB input capture source, and the widths of 2-phase encoder 4-multiplication pulses are detected.

MTU1.TGRA and MTU1.TGRB for MTU1 are designated for the input capture function and MTU0.TGRA and MTU0.TGRC compare matches in MTU0 are selected as the input capture sources to store the up-counter/down-counter values for the control cycles.

This procedure enables the accurate detection of position and speed.

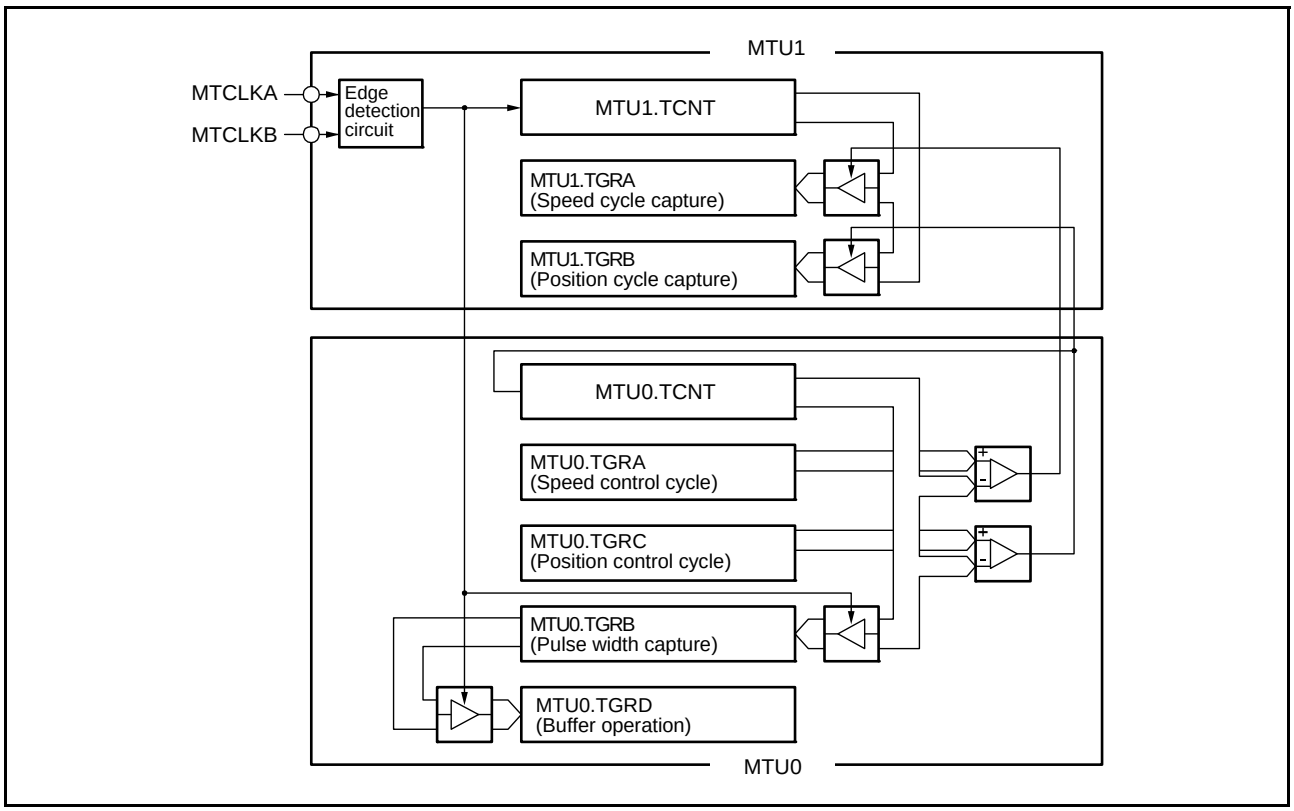



**Figure 20.34 Phase Counting Mode Application Example**

### 20.3.7 Reset-Synchronized PWM Mode

In the reset-synchronized PWM mode, three phases of positive and negative PWM waveforms that share a common wave transition point can be output by combining MTU3 and MTU4.
When set for reset-synchronized PWM mode, the MTIOC3B, MTIOC3D, MTIOC4A, MTIOC4C, MTIOC4B, and MTIOC4D pins function as PWM output pins and timer counter 3 (MTU3.TCNT) functions as an up-counter.
Table 20.52 lists the PWM output pins. Table 20.53 lists the settings of the registers.

**Table 20.52 Output Pins for Reset-Synchronized PWM Mode**

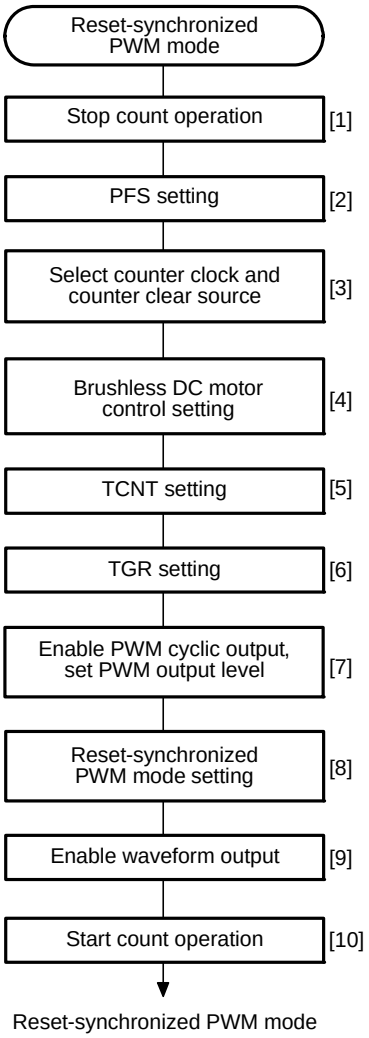
| Channel | Output Pin | Description |
|---|---|---|
| MTU3 | MTIOC3B | PWM output pin 1 |
| | MTIOC3D | PWM output pin 1' (negative-phase waveform of PWM output 1) |
| MTU4 | MTIOC4A | PWM output pin 2 |
| | MTIOC4C | PWM output pin 2' (negative-phase waveform of PWM output 2) |
| | MTIOC4B | PWM output pin 3 |
| | MTIOC4D | PWM output pin 3' (negative-phase waveform of PWM output 3) |

**Table 20.53 Register Settings for Reset-Synchronized PWM Mode**

| Register | Setting |
|---|---|
| MTU3.TCNT | Initial setting (0000h) |
| MTU4.TCNT | Initial setting (0000h) |
| MTU3.TGRA | Set the count cycle for MTU3.TCNT |
| MTU3.TGRB | Set the transition point of the PWM waveform to be output from the MTIOC3B and MTIOC3D pins |
| MTU4.TGRA | Set the transition point of the PWM waveform to be output from the MTIOC4A and MTIOC4C pins |
| MTU4.TGRB | Set the transition point of the PWM waveform to be output from the MTIOC4B and MTIOC4D pins |

### (1) Example of Procedure for Setting Reset-Synchronized PWM Mode

Figure 20.35 shows an example of procedure for setting the reset-synchronized PWM mode.




[1] Clear the CST3 and CST4 bits in TSTR to 0 to stop the TCNT operation. Specify the reset-synchronized PWM mode while MTU3.TCNT and MTU4.TCNT are stopped.

[2] Set the Pmn pin function control register and the port I/O register.

[3] Set bits TPSC[2:0] and CKEG[1:0] in the MTU3.TCR to select the counter clock and clock edge for channel 3. Set the CCLR[2:0] bits in the MTU3.TCR to select TGRA compare-match as a counter clear source.

[4] When performing brushless DC motor control, set the BDC bit in the timer gate control register (TGCR) and set the feedback signal input source and output chopping or gate signal direct output

[5] Set MTU3.TCNT and MTU4.TCNT to 0000h.

[6] MTU3.TGRA is the cycle register. Set the waveform cycle value in MTU3.TGRA. Set the transition timing of the PWM output waveforms in MTU3.TGRB, MTU4.TGRA, and MTU4.TGRB. Set times within the compare match range of MTU3.TCNT.

$X \leq$ MTU3.TGRA (X: Setting value)

[7] Enable or disable toggle output synchronized with the PWM cycle using the PSYE bit in TOCR1, and set the PWM output level with bits OLSP and OLSN. When specifying the PWM output Level. In buffer operation of TOCR2 and TOLBR, see Figure 21.3.

[8] Set the MD[3:0] bits in the timer mode register (MTU3.TMDR) to 1000b to select the reset-synchronized PWM mode. Set the BFA bit to 1 for buffer operation of TGRA and TGRC, and the BFB bit to 1 for buffer operation of TGRB and TGRD, if necessary. Set the initial value in MTU4.TMDR.

[9] Set the enabling/disabling of the PWM waveform output pin in TOER.

[10] Set the CST3 bit in the TSTR to 1 to start the count operation.

Note: • The output waveform starts to toggle at the point of MTU3.TCNT = MTU3.TGRA = X by setting X = MTU3.TGRA, i.e., cycle = duty cycle.

**Figure 20.35 Procedure for Selecting Reset-Synchronized PWM Mode**

### (2) Example of Reset-Synchronized PWM Mode Operation

Figure 20.36 shows an example of operation in the reset-synchronized PWM mode.
MTU3.TCNT and MTU4.TCNT operate as up-counters. The counters are cleared when a compare match occurs between MTU3.TCNT and MTU3.TGRA, and then begin incrementing from 0000h. The output from the PWM pins toggles every time a compare match occurs in MTU3.TGRB, MTU4.TGRA, and MTU4.TGRB and the counters are cleared.

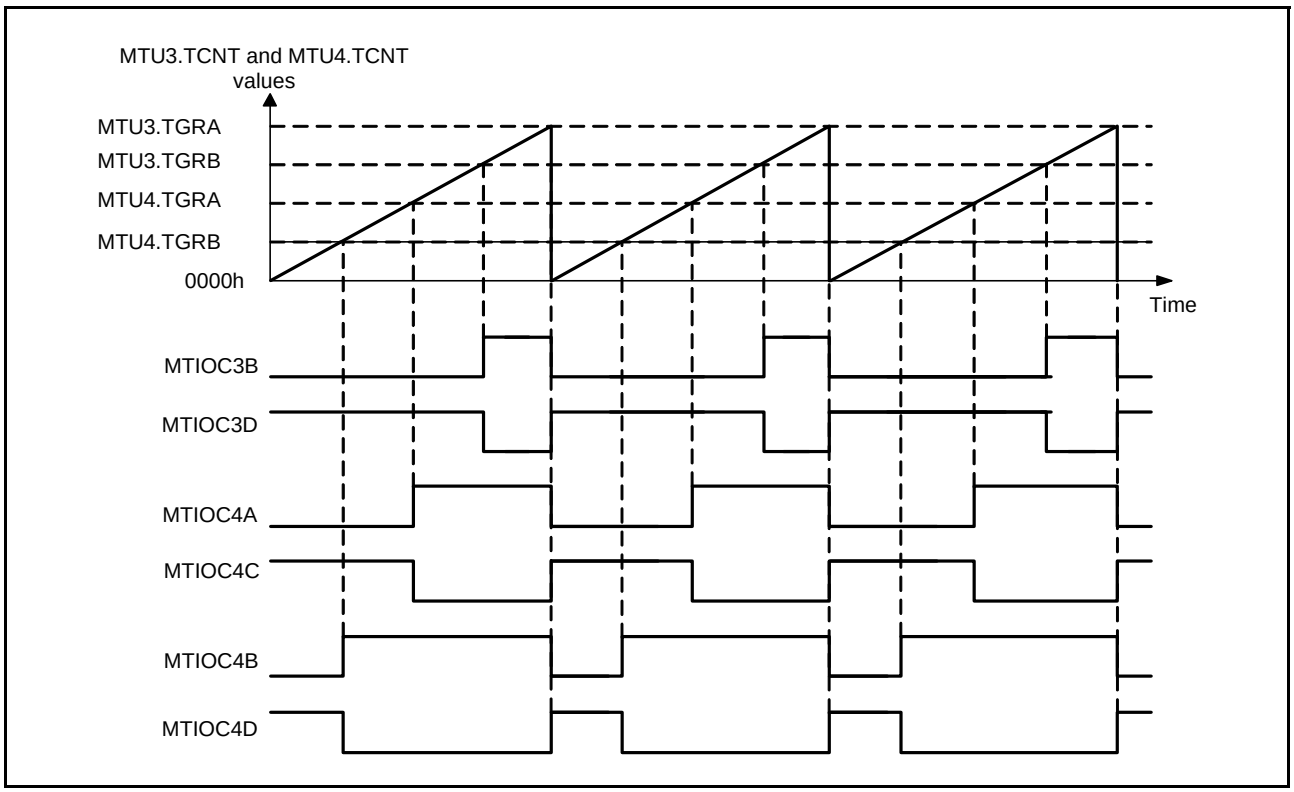



**Figure 20.36 Example of Reset-Synchronized PWM Mode Operation (When TOCR1.OLSN = 1 and OLSP = 1)**

### 20.3.8 Complementary PWM Mode

In complementary PWM mode, dead time can be set for PWM waveforms to be output. The dead time is the period during which the upper and lower arm transistors are set to the inactive level in order to prevent short-circuiting of the arms. Six 3-phase PWM waveforms can be output by using MTU3 and MTU4. PWM waveforms without dead time can also be output.

In complementary PWM mode, MTIOC3B, MTIOC3D, MTIOC4A, MTIOC4B, MTIOC4C, and MTIOC4D pins function as PWM output pins, and the MTIOC3A pin can be set for toggle output synchronized with the PWM cycle. MTU3.TCNT and MTU4.TCNT function as up/down-counters.

Table 20.54 lists the PWM output pins used. Table 20.55 lists the settings of the registers used.

A function to directly cut off the PWM output by using an external signal is supported as a port function.

**Table 20.54 Output Pins for Complementary PWM Mode**

| Channel | Output Pin | Description |
|---|---|---|
| MTU3 | MTIOC3A | Toggle output synchronized with PWM cycle (or I/O port) |
| | MTIOC3B | PWM output pin 1 |
| | MTIOC3C | I/O port*1 |
| | MTIOC3D | PWM output pin 1' (Negative-phase waveform output of PWM output 1) |
| MTU4 | MTIOC4A | PWM output pin 2 |
| | MTIOC4C | PWM output pin 2' (Negative-phase waveform output of PWM output 2) |
| | MTIOC4B | PWM output pin 3 |
| | MTIOC4D | PWM output pin 3' (Negative-phase waveform output of PWM output 3) |

Note 1. Avoid setting the MTIOC3C pin as a timer I/O pin in complementary PWM mode.

**Table 20.55 Register Settings for Complementary PWM Mode**

| Channel | Counter/ Register | Description | Read/Write from CPU |
|---|---|---|---|
| MTU3 | MTU3.TCNT | Starts up-counting from the value set in the dead time register | Maskable by TRWER setting*1 |
| | MTU3.TGRA | Set MTU3.TCNT upper limit value (1/2 carrier cycle + dead time) | Maskable by TRWER setting*1 |
| | MTU3.TGRB | PWM output 1 compare register | Maskable by TRWER setting*1 |
| | MTU3.TGRC | MTU3.TGRA buffer register | Readable/writable |
| | MTU3.TGRD | PWM output 1/MTU3.TGRB buffer register | Readable/writable |
| MTU4 | MTU4.TCNT | Starts up-counting after being initialized to 0000h | Maskable by TRWER setting*1 |
| | MTU4.TGRA | PWM output 2 compare register | Maskable by TRWER setting*1 |
| | MTU4.TGRB | PWM output 3 compare register | Maskable by TRWER setting*1 |
| | MTU4.TGRC | PWM output 2/MTU4.TGRA buffer register | Readable/writable |
| | MTU4.TGRD | PWM output 3/MTU4.TGRB buffer register | Readable/writable |
| Timer dead time data register (TDDR) | | Set MTU4.TCNT and MTU3.TCNT offset value (dead time value) | Maskable by TRWER setting*1 |
| Timer cycle data register (TCDR) | | Set MTU4.TCNT upper limit value (1/2 carrier cycle) | Maskable by TRWER setting*1 |
| Timer cycle buffer register (TCBR) | | TCDR buffer register | Readable/writable |
| Subcounter (TCNTS) | | Subcounter for dead time generation | Read-only |
| Temporary register 1 (TEMP1) | | PWM output 1/MTU3.TGRB temporary register | Not readable/writable |
| Temporary register 2 (TEMP2) | | PWM output 2/MTU4.TGRA temporary register | Not readable/writable |
| Temporary register 3 (TEMP3) | | PWM output 3/MTU4.TGRB temporary register | Not readable/writable |

Note 1. Access can be enabled or disabled according to the setting in TRWER (timer read/write enable register).

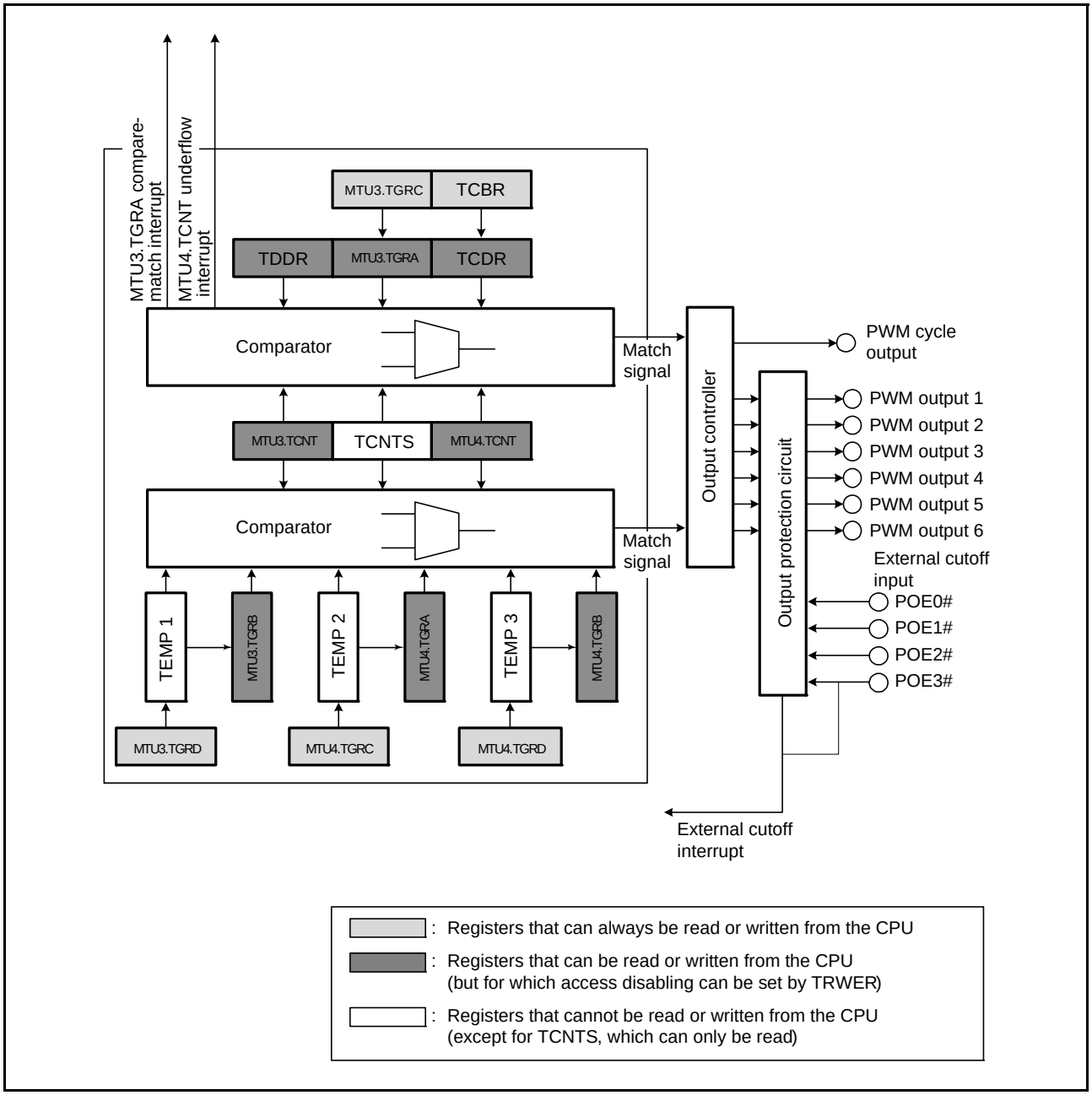
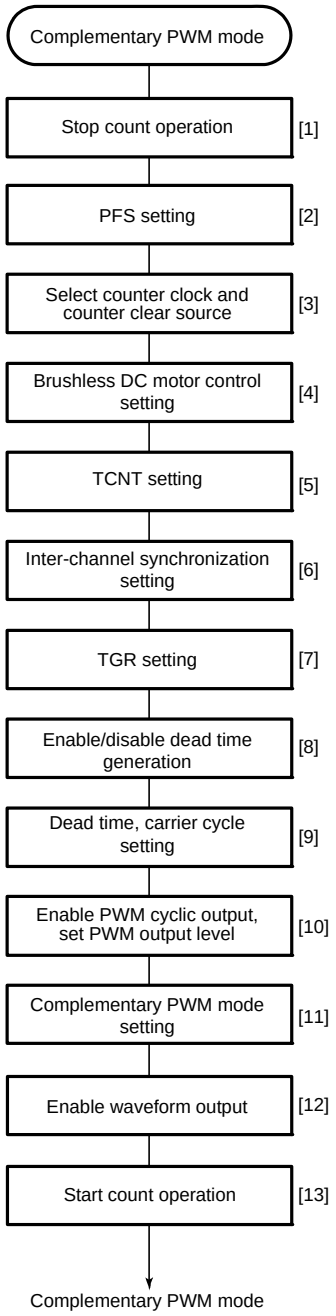



**Figure 20.37 Block Diagram of MTU3 and MTU4 in Complementary PWM Mode**

## (1) Example of Complementary PWM Mode Setting Procedure

Figure 20.38 shows an example of the complementary PWM mode setting procedure.




[1] Clear bits CST3 and CST4 in the timer start register (TSTR) to 0 to stop timer counter (TCNT) operation. Specify complementary PWM mode while MTU3.TCNT and MTU4.TCNT are stopped.

[2] Set the Pmn pin function control register and the port I/O register.

[3] Set the same counter clock and clock edge for MTU3 and MTU4 with bits TPSC[2:0] and bits CKEG[1:0] in the timer control register (TCR). Use the CCLR[2:0] bits to set synchronous clearing only when restarting by synchronous clearing with another channel during complementary PWM mode operation.

[4] When performing brushless DC motor control, set bit BDC in the timer gate control register (TGCR) and set the feedback signal input source and output chopping or gate signal direct output.

[5] Set the dead time in MTU3.TCNT. Set MTU4.TCNT to 0000h.

[6] Set only when restarting by synchronous clearing with another channel during complementary PWM mode operation. In this case, synchronize the channel generating the synchronous clear with MTU3 and MTU4 using the timer synchronous register (TSYR).

[7] Set the output PWM duty cycles in the compare registers (MTU3.TGRB, MTU4.TGRA, and MTU4.TGRB) and buffer registers (MTU3.TGRD, MTU4.TGRC, and MTU4.TGRD). Set the same value in the buffer registers and the compare registers.

[8] This setting is necessary only when no dead time should be generated. Make appropriate settings in the timer dead time enable register (TDER) so that no dead time is generated.

[9] Set the dead time in the dead time register (TDDR), 1/2 the carrier cycle in the timer cycle data register (TCDR) and timer cycle buffer register (TCBR), and 1/2 the carrier cycle plus the dead time in MTU3.TGRA and MTU3.TGRC. When no dead time generation is selected, set 1 in TDDR and 1/2 the carrier cycle + 1 in MTU3.TGRA and MTU3.TGRC.

[10] Enable or disable toggle output synchronized with the PWM cycle using bit PSYE in the timer output control register 1 (TOCR1), and set the PWM output level with bits OLSP and OLSN.
When specifying the PWM output level by using TOLBR as a buffer for MTU.TOCR2, see Figure 21.3, Example of PWM Output Level Setting Procedure in Buffer Operation.

[11] Select complementary PWM mode in the timer mode register (MTU3.TMDR). Set the BFA bit in TGRA and TGRC to 1 for buffer operation of TGRA and TGRC, and the BFB bit to 1 for buffer operation of TGRB and TGRD, if necessary. Set nothing in MTU4.TMDR.

[12] Set enabling/disabling of PWM waveform output pin output in the timer output master enable register (TOER).

[13] Set bits CST3 and CST4 in TSTR to 1 simultaneously to start the count operation.

**Figure 20.38 Example of Complementary PWM Mode Setting Procedure**

### (2) Outline of Complementary PWM Mode Operation

In complementary PWM mode, six 3-phase PWM waveforms of PWM waveforms can be output. Figure 20.39 illustrates counter operation in complementary PWM mode, and Figure 20.40 shows an example of operation in complementary PWM mode.

#### (a) Counter Operation

In complementary PWM mode, three counters—MTU3.TCNT, MTU4.TCNT, and TCNTS— perform up-/down-count operations.

MTU3.TCNT is automatically initialized to the value set in TDDR when complementary PWM mode is selected and the CST bit in TSTR is 0.

When the CST bit is set to 1, MTU3.TCNT counts up to the value set in MTU3.TGRA, then switches to down-counting when it matches MTU3.TGRA. When the MTU3.TCNT value matches TDDR, the counter switches to up-counting, and the operation is repeated in this way.

MTU4.TCNT should be initialized to 0000h.

When the CST bit is set to 1, MTU4.TCNT counts up in synchronization with MTU3.TCNT, and switches to down-counting when it matches TCDR. On reaching 0000h, MTU4.TCNT switches to up-counting, and the operation is repeated in this way.

TCNTS is a read-only counter. It does not need to be initialized.

When MTU3.TCNT matches TCDR during up-/down-counting of TCNT in MTU3 and MTU4, TCNTS starts down-counting, and when TCNTS matches TCDR, the operation switches to up-counting. When TCNTS matches MTU3.TGRA, it is cleared to 0000h.

When MTU4.TCNT matches TDDR during down-counting of MTU3.TCNT and MTU4.TCNT, TCNTS starts up-counting, and when TCNTS matches TDDR, the operation switches to down-counting. When TCNTS reaches 0000h, it is set with the value in MTU3.TGRA.

TCNTS is compared with the compare register and temporary register, in which the PWM duty is specified, only during the count operation.

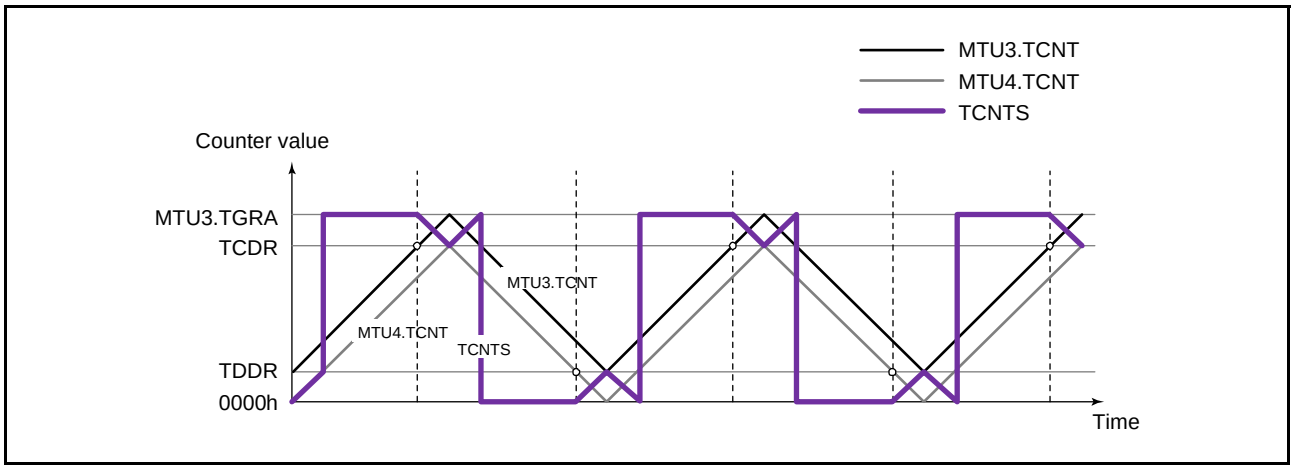



**Figure 20.39 Counter Operation in Complementary PWM Mode**

### (b) Register Operation

In complementary PWM mode, nine registers (compare registers, buffer registers, and temporary registers) are used. Figure 20.40 shows an example of operation in complementary PWM mode.

MTU3.TGRB, MTU4.TGRA, and MTU4.TGRB are constantly compared with the counters to generate PWM waveforms. When these registers match the counter, the value set in bits OLSN and OLSP in the timer output control register 1 (TOCR1) is output.

MTU3.TGRD, MTU4.TGRC, and MTU4.TGRD are buffer registers for these compare registers. Between a buffer register and a compare register, there is a temporary register. The temporary registers cannot be accessed by the CPU. Data in a compare register can be changed by writing new data to the corresponding buffer register. The buffer registers can be read or written at any time.

The data written to a buffer register is constantly transferred to the temporary register in the Ta interval. Data is not transferred to the temporary register in the Tb interval. Data written to a buffer register in this interval is transferred to the temporary register at the end of the Tb interval.

The value transferred to a temporary register is transferred to the compare register when TCNTS for which the Tb interval ends matches MTU3.TGRA while TCNTS is counting up, or 0000h while counting down. The timing for transfer from the temporary register to the compare register can be selected with bits MD[3:0] in the timer mode register (TMDR). Figure 20.40 shows an example in which the trough is selected for the transfer timing.

In the Tb (Tb2 in Figure 20.40) interval in which data is not transferred to the temporary register, the temporary register has the same function as the compare register and is compared with the counter. In this interval, therefore, there are two compare match registers for one output phase; the compare register contains the pre-change data and the temporary register contains new data. In this interval, three counters (MTU3.TCNT, MTU4.TCNT and TCNTS) and two registers (compare register and temporary register) are compared, and PWM output is controlled accordingly.

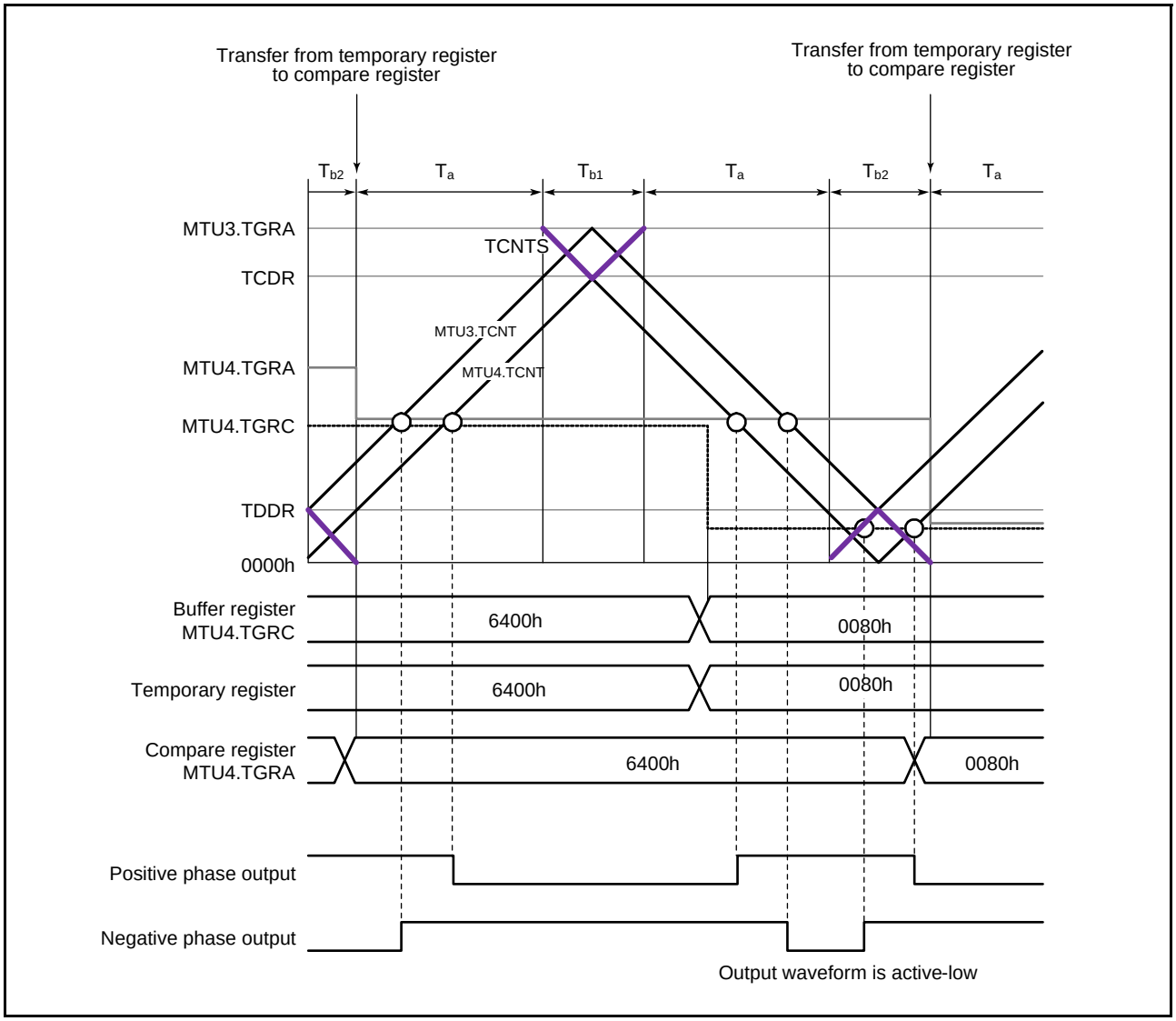



**Figure 20.40 Example of Operation in Complementary PWM Mode**

### (c) Initial Setting

In complementary PWM mode, there are six registers that require initial setting. In addition, there is a register that specifies whether to generate dead time (it should be used only when dead time generation should be disabled).
Before setting complementary PWM mode with bits MD[3:0] in the timer mode register (TMDR), initial values should be set in the following registers.
MTU3.TGRC operates as the buffer register for MTU3.TGRA, and should be set with 1/2 the PWM carrier cycle + dead time Td. The timer cycle buffer register (TCBR) operates as the buffer register for the timer cycle data register (TCDR), and should be set with 1/2 the PWM carrier cycle. Set dead time Td in the timer dead time data register (TDDR).

When dead time is not needed, the TDER bit in the timer dead time enable register (TDER) should be cleared to 0, MTU3.TGRC and MTU3.TGRA should be set to 1/2 the PWM carrier cycle + 1, and TDDR should be set to 1.
Set the respective initial PWM duty values in three buffer registers MTU3.TGRD, MTU4.TGRC, and MTU4.TGRD. The values set in the five buffer registers excluding TDDR are transferred to the corresponding compare registers as soon as complementary PWM mode is set.
Set MTU4.TCNT to 0000h before setting complementary PWM mode.

**Table 20.56 Registers and Counters Requiring Initial Setting**

| Register and Counter | Setting |
|---|---|
| MTU3.TGRC | 1/2 PWM carrier cycle + dead time Td<br>(1/2 PWM carrier cycle + 1 when dead time generation is disabled by TDER) |
| TDDR | Dead time Td (1 when dead time generation is disabled by TDER) |
| TCBR | 1/2 PWM carrier cycle |
| MTU3.TGRD,<br>MTU4.TGRC,<br>MTU4.TGRD | Initial PWM duty value for each phase |
| MTU4.TCNT | 0000h |

Note: • The value set in MTU3.TGRC should be the sum of 1/2 the PWM carrier cycle set in TCBR and dead time Td set in TDDR. When dead time generation is disabled by TDER, TGRC should be set to 1/2 the PWM carrier cycle + 1.

### (d) PWM Output Level Setting

In complementary PWM mode, the PWM pulse output level is set with bits OLSN and OLSP in timer output control register 1 (TOCR1) or bits OLS1P to OLS3P and OLS1N to OLS3N in timer output control register 2 (TOCR2).
The output level can be set for each of the three positive phases and three negative phases of 6-phase output.
Complementary PWM mode should be cleared before setting or changing output levels.

### (e) Dead Time Setting

In complementary PWM mode, dead time can be set for PWM waveforms to be output.
The dead time is set in the timer dead time data register (TDDR). The value set in TDDR is used as the MTU3.TCNT counter start value and creates a dead time between MTU3.TCNT and MTU4.TCNT. Complementary PWM mode should be cleared before changing the contents of TDDR.

### (f) Dead Time Suppressing

Dead time generation is suppressed by clearing the TDER bit in the timer dead time enable register (TDER) to 0. TDER bit can be cleared to 0 only when 0 is written to it after reading it as 1.
MTU3.TGRA and MTU3.TGRC should be set to 1/2 PWM carrier cycle + 1 and the timer dead time data register (TDDR) should be set to 1.
By the above settings, PWM waveforms without dead time can be obtained. Figure 20.41 shows an example of operation without dead time.

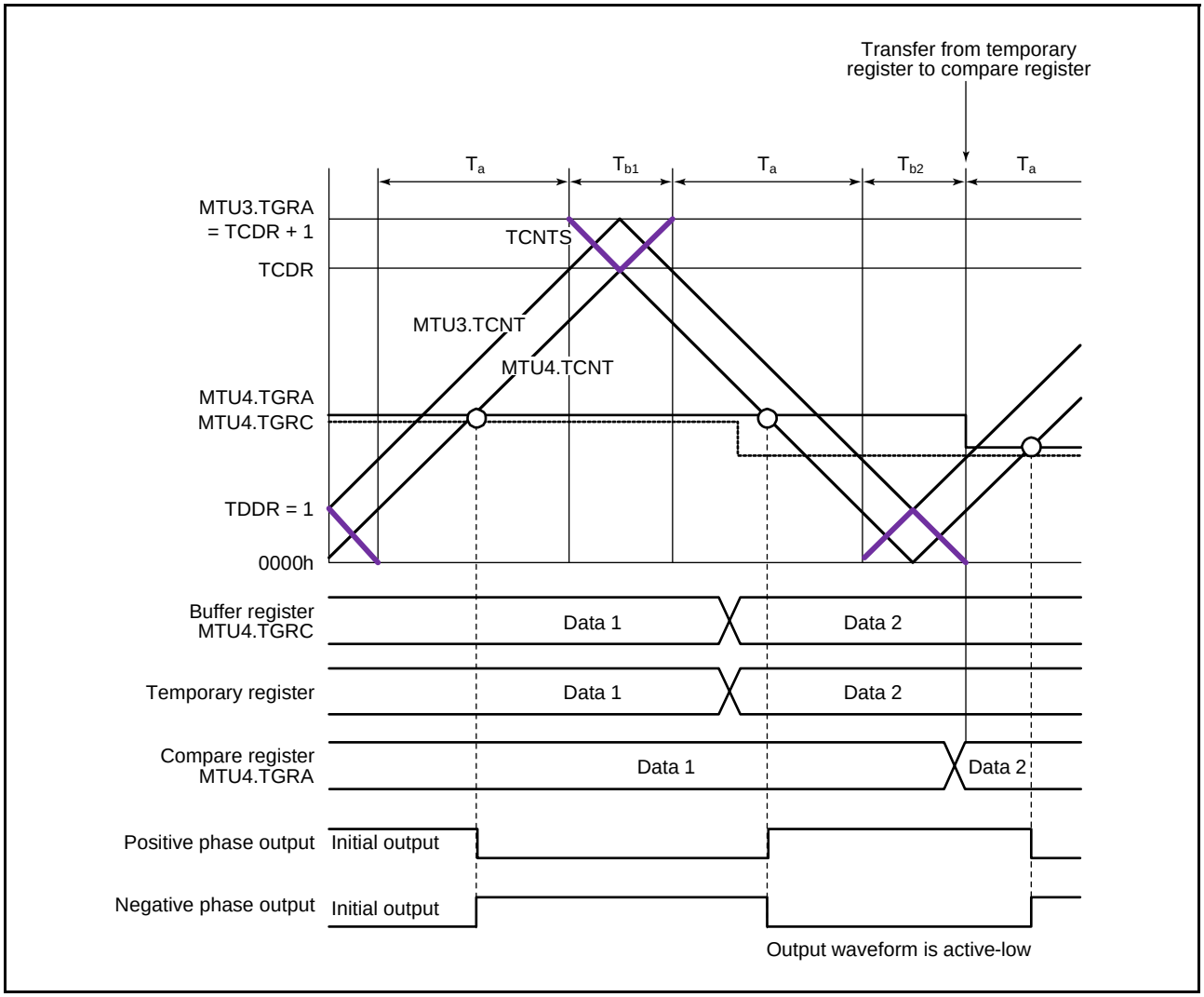



**Figure 20.41 Example of Operation without Dead Time**

### (g) PWM Cycle Setting

In complementary PWM mode, the PWM pulse cycle is set in two registers—MTU3.TGRA, in which the MTU3.TCNT upper limit value is set, and TCDR, in which the MTU4.TCNT upper limit value is set. The settings should be made so as to achieve the following relationship between these two registers:

With dead time: MTU3.TGRA setting = TCDR setting + TDDR setting
Without dead time: MTU3.TGRA setting = TCDR setting + 1

The settings should be made so as to achieve the following relationship between registers TCDR and TDDR.
TCDR setting > TDDR setting × 2 + 2

The MTU3.TGRA and MTU3.TCDR settings are made by setting values in buffer registers MTU3.TGRC and MTU3.TCBR. The values set in MTU3.TGRC and MTU3.TCBR are transferred simultaneously to MTU3.TGRA and MTU3.TCDR with the transfer timing selected with the TMDR.MD[3:0] bits.
The new PWM cycle is reflected from the next cycle when data is updated at the crest, or from the current cycle when updated in the trough. Figure 20.42 illustrates the operation when the PWM cycle is updated at the crest.
Refer to the following section (h), Register Data Updating, for the method of updating the data in each buffer register.

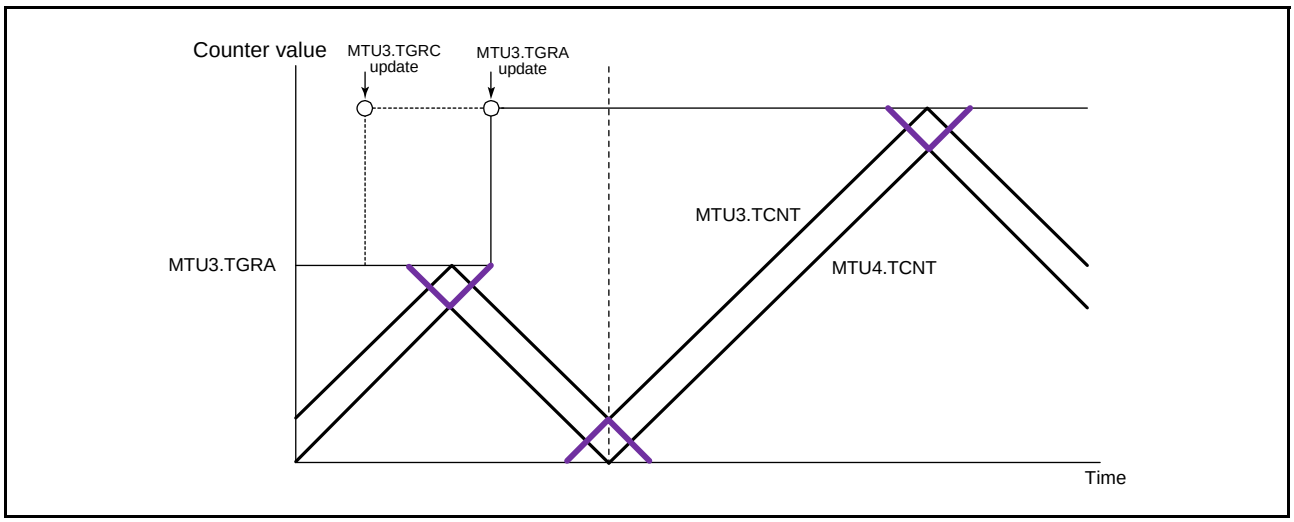



**Figure 20.42 Example of PWM Cycle Updating**

### (h) Register Data Updating

In complementary PWM mode, the buffer register is used to update the data in a compare register. The update data can be written to the buffer register at any time. There are five registers (PWM duty and carrier cycle registers) that have buffer registers and can be updated during operation.
There is a temporary register between each of these registers and its buffer register. While subcounter TCNTS is not counting, if buffer register data is updated, the temporary register value also changes. Data is not transferred from buffer registers to temporary registers while TCNTS is counting; in this case, the value written to a buffer register is transferred after TCNTS halts.
The temporary register value is transferred to the compare register at the data update timing set with the TMDR.MD[3:0] bits. Figure 20.43 shows an example of data updating in complementary PWM mode. This example shows the mode in which data is updated at both the counter crest and trough.
When updating buffer register data, be sure to write to MTU4.TGRD at the end of the update. Data is transferred from buffer registers to the temporary registers simultaneously for all five registers after the write to MTU4.TGRD.
Even when not updating all five registers or when not updating the MTU4.TGRD data, be sure to write to MTU4.TGRD after writing data to the registers to be updated. In this case, the data written to MTU4.TGRD should be the same as the

data prior to the write operation.

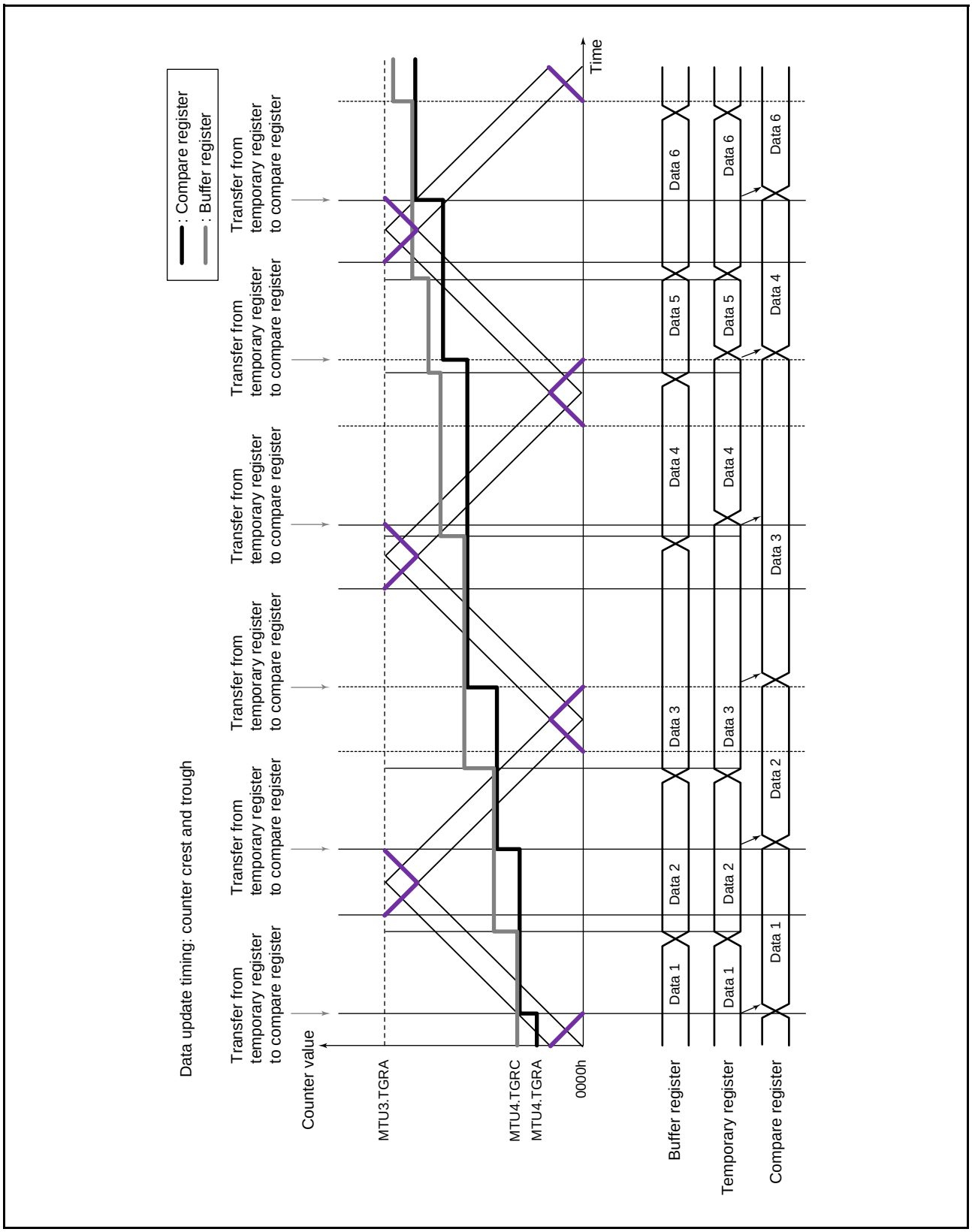



**Figure 20.43 Example of Data Updating in Complementary PWM Mode**

### (i) Initial Output in Complementary PWM Mode

In complementary PWM mode, the initial output is determined by the setting of bits OLSN and OLSP in timer output control register 1 (TOCR1) or bits OLS1N to OLS3N and OLS1P to OLS3P in timer output control register 2 (TOCR2). This initial output is the non-active level of the PWM pulse and continues from when complementary PWM mode is set with the timer mode register (TMDR) until MTU4.TCNT exceeds the value set in the dead time register (TDDR). Figure 20.44 shows an example of the initial output in complementary PWM mode.
An example of the waveform when the initial PWM duty value is smaller than the TDDR value is shown in Figure 20.45.

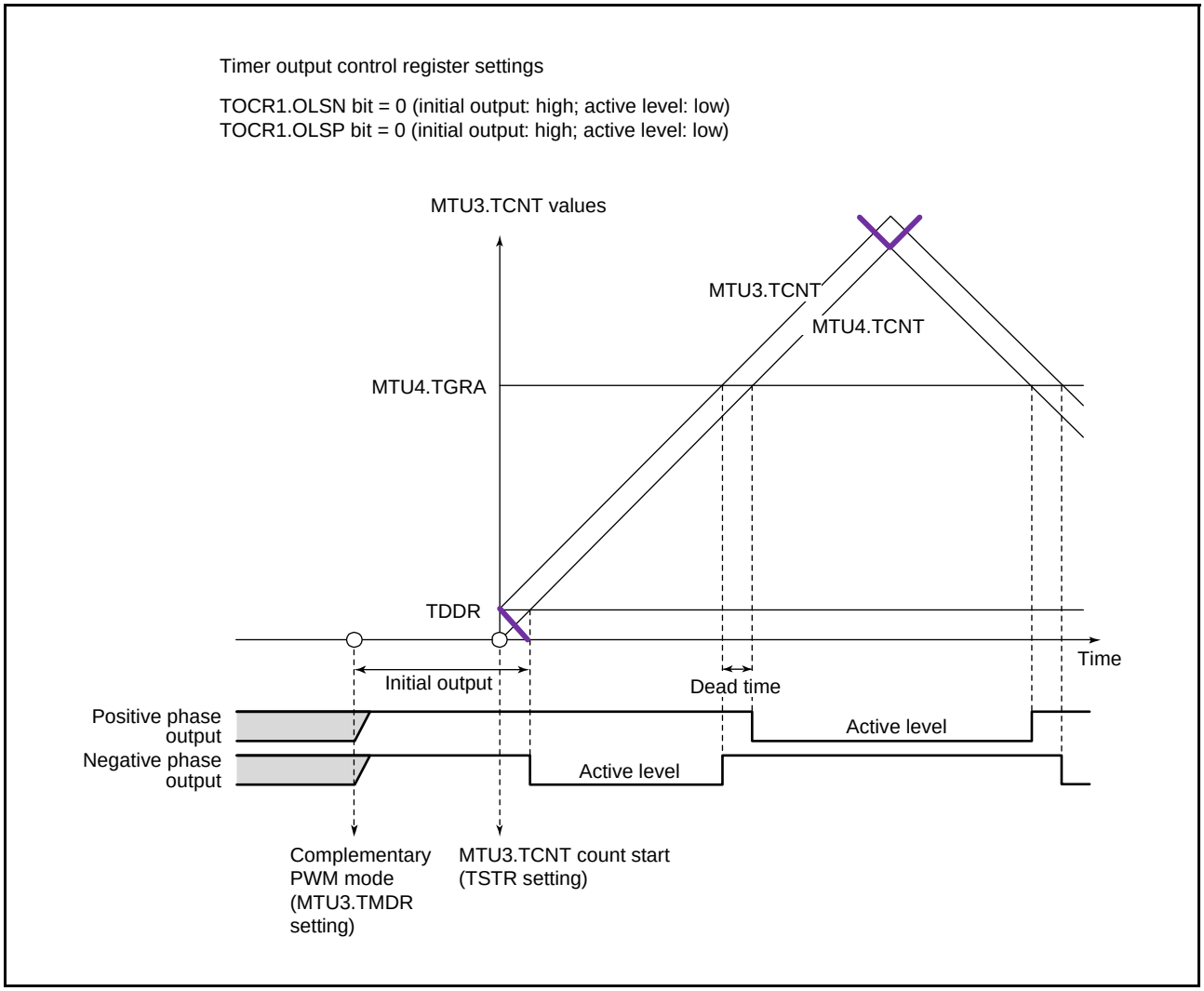



**Figure 20.44 Example of Initial Output in Complementary PWM Mode (1)**

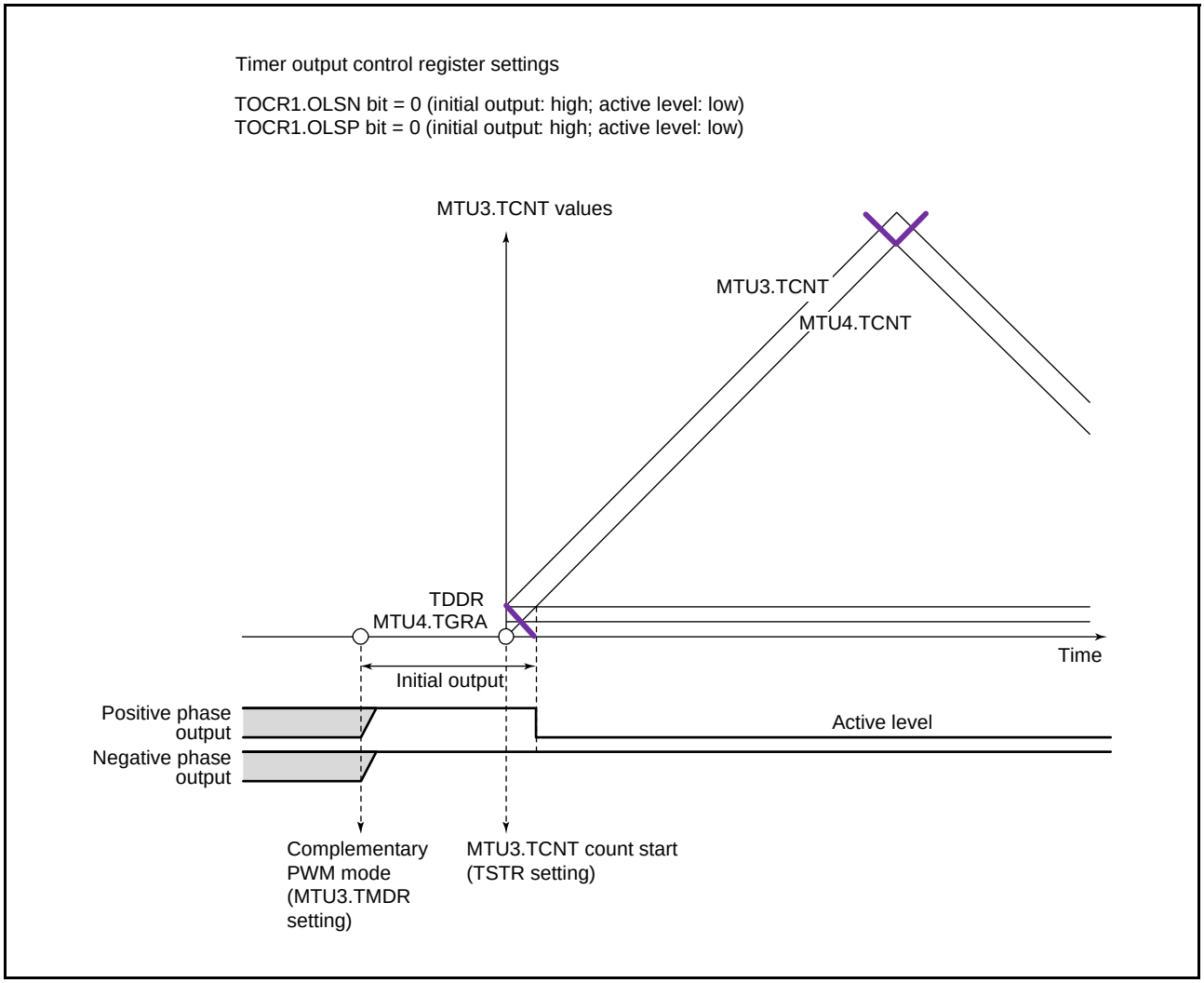



**Figure 20.45 Example of Initial Output in Complementary PWM Mode (2)**

### (j) Method for Generating PWM Output in Complementary PWM Mode

In complementary PWM mode, six 3-phase of PWM waveforms can be output. Dead time can be set for PWM waveforms to be output.

A PWM waveform is generated by output of the level selected in the timer output control register in the event of a compare match between a counter and a compare register. While TCNTS is counting, the compare register and temporary register values are simultaneously compared to create consecutive PWM pulses from 0% to 100%. The relative timing of turn-on and turn-off compare match occurrence may vary, but the compare match that turns off each phase takes precedence to secure the dead time and ensure that the positive-phase and negative-phase turn-on times do not overlap. Figure 20.46 to Figure 20.48 show examples of waveform generation in complementary PWM mode.

The positive-phase and negative-phase turn-off timing is generated by a compare match with the MTU3.TCNT counter, and the turn-on timing by a compare match with the MTU4.TCNT counter, which operates with a delay of the dead time behind the MTU3.TCNT counter. In the T1 period, compare match a that turns off the negative phase has the highest priority, and compare matches before a are ignored. In the T2 period, compare match c that turns off the positive phase has the highest priority, and compare matches before c are ignored.

In most cases, compare matches occur in the order a → b → c → d (or c → d → a' → b') as shown in Figure 20.46.

If compare matches deviate from the a → b → c → d order, since the time for which the negative phase is off is shorter than twice the dead time, the positive phase is not turned on. If compare matches deviate from the c → d → a' → b' order, since the time for which the positive phase is off is shorter than twice the dead time, the negative phase is not turned on.

As shown in Figure 20.47, if compare match c follows compare match a before compare match b, compare match b is ignored and the negative phase is turned on by compare match d. This is because turning off the positive phase has higher priority due to the occurrence of compare match c (positive-phase off timing) before compare match b (positive-phase on timing) (consequently, the waveform does not change because the positive phase goes from off to off).

Similarly, in the example in Figure 20.48, compare match a' with new data in the temporary register occurs before compare match c, but until compare match c, which turns off the positive phase, other compare matches are ignored. As a result, the negative phase is not turned on.

Thus, in complementary PWM mode, compare matches at turn-off timings take precedence, and turn-on timing compare matches that occur before a turn-off timing compare match are ignored.

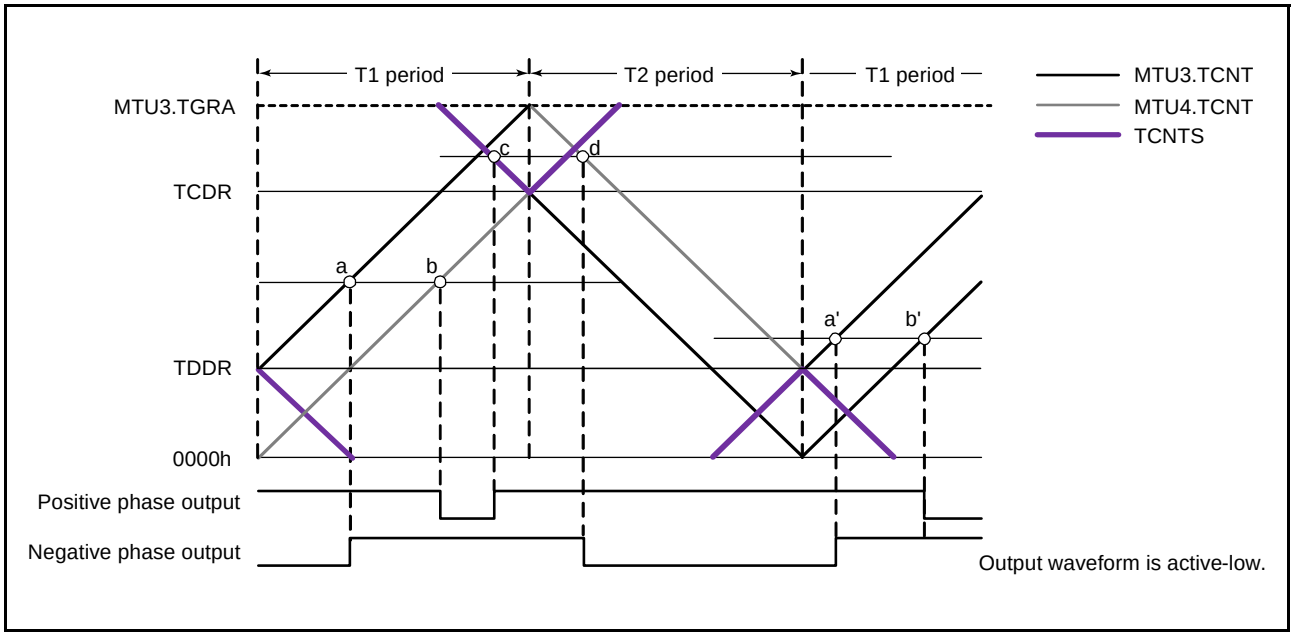



**Figure 20.46 Example of Waveform Output in Complementary PWM Mode (1)**

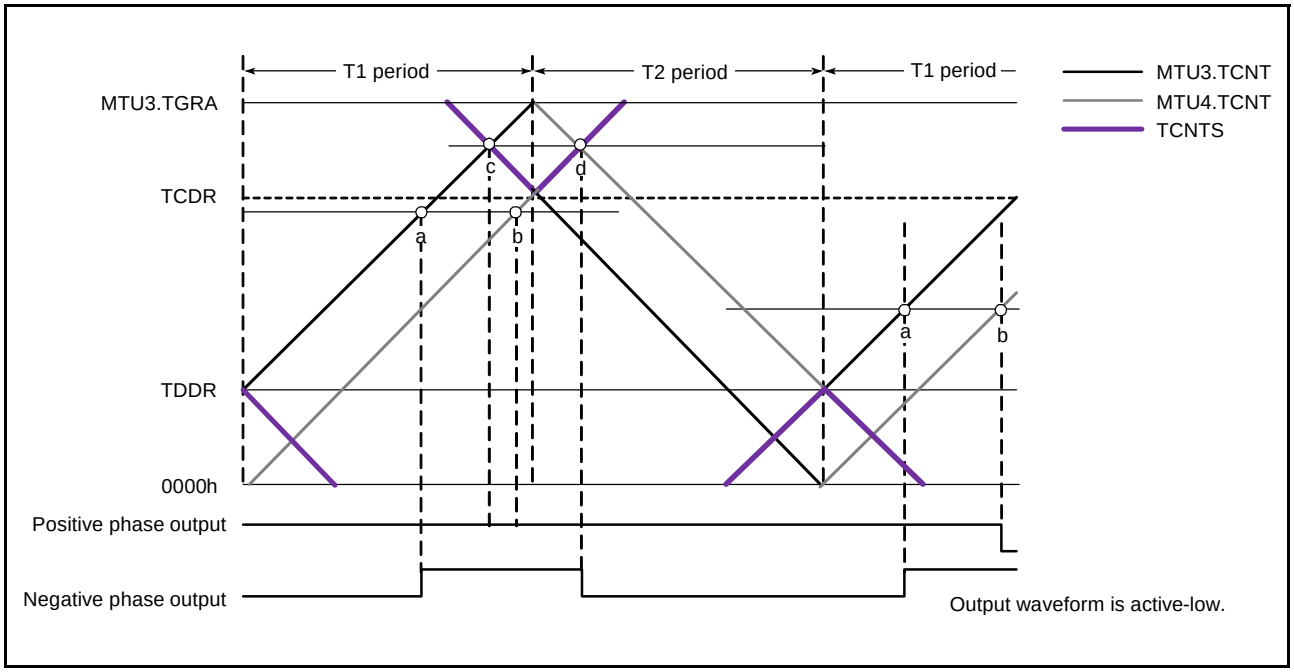



**Figure 20.47 Example of Waveform Output in Complementary PWM Mode (2)**

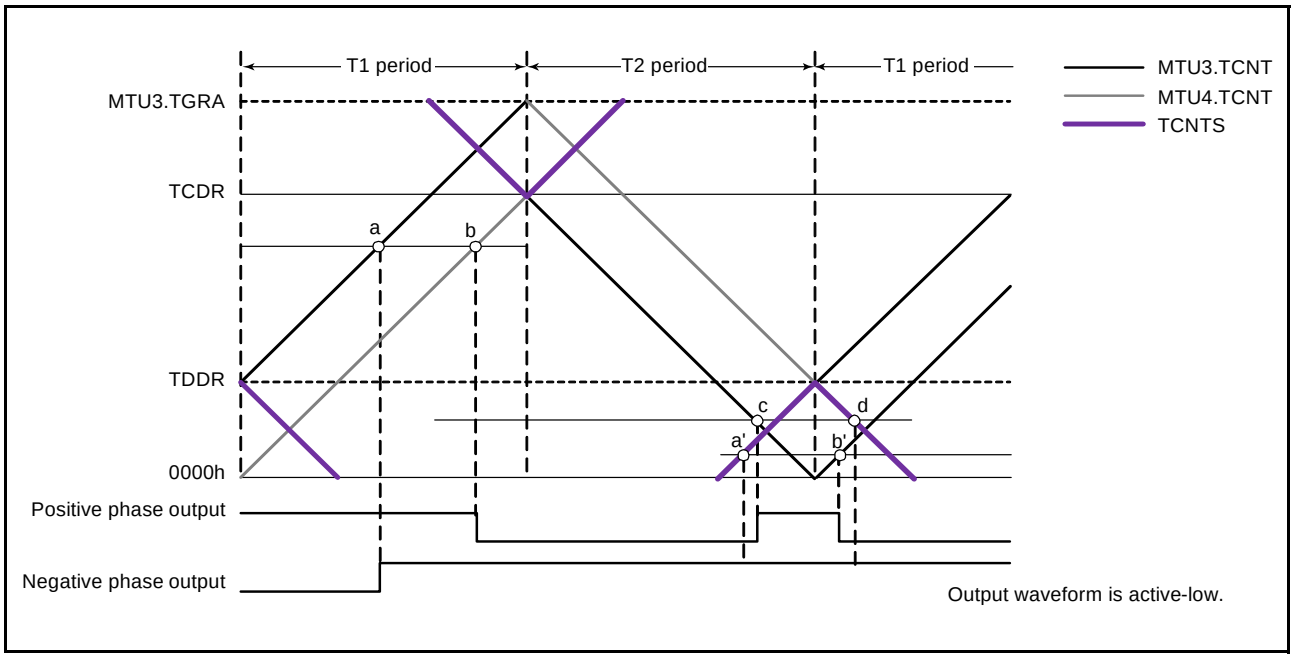



**Figure 20.48 Example of Waveform Output in Complementary PWM Mode (3)**

### (k) 0% and 100% Duty Cycle Output in Complementary PWM Mode

In complementary PWM mode, 0% and 100% duty cycle PWM waveforms can be output as required. Figure 20.49 to Figure 20.53 show output examples.

A 100% duty cycle waveform is output when the data register value is set to 0000h. The waveform in this case has a positive phase with a 100% on-state. A 0% duty cycle waveform is output when the data register value is set to the same value as MTU3.TGRA. The waveform in this case has a positive phase with a 100% off-state.

On and off compare matches occur simultaneously, but if a turn-on compare match and turn-off compare match for the same phase occur simultaneously, both compare matches are ignored and the waveform does not change.

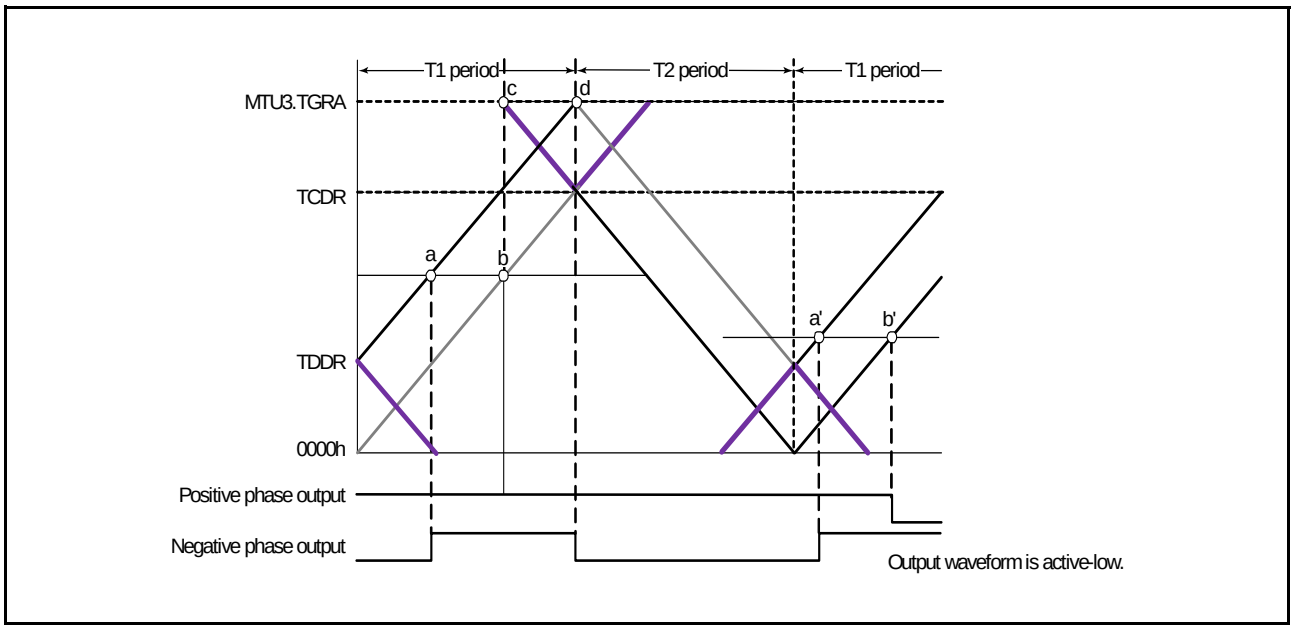



**Figure 20.49 Example of 0% and 100% Waveform Output in Complementary PWM Mode (1)**

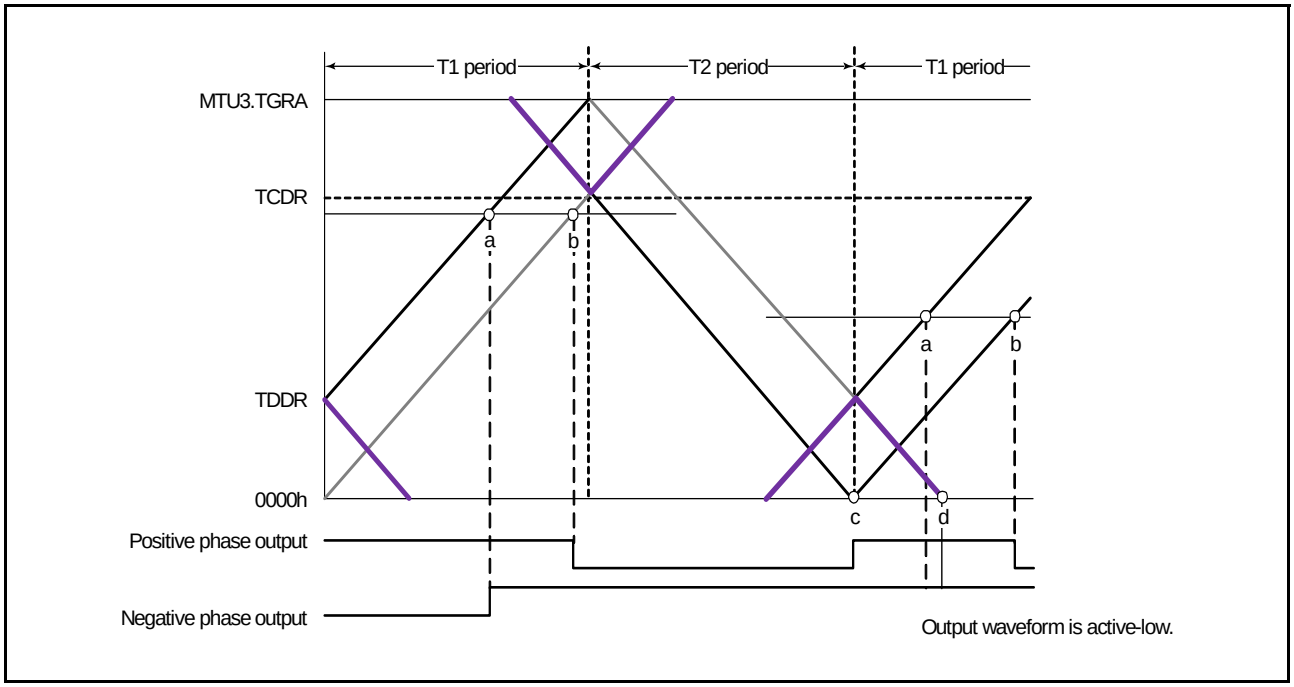



**Figure 20.50 Example of 0% and 100% Waveform Output in Complementary PWM Mode (2)**

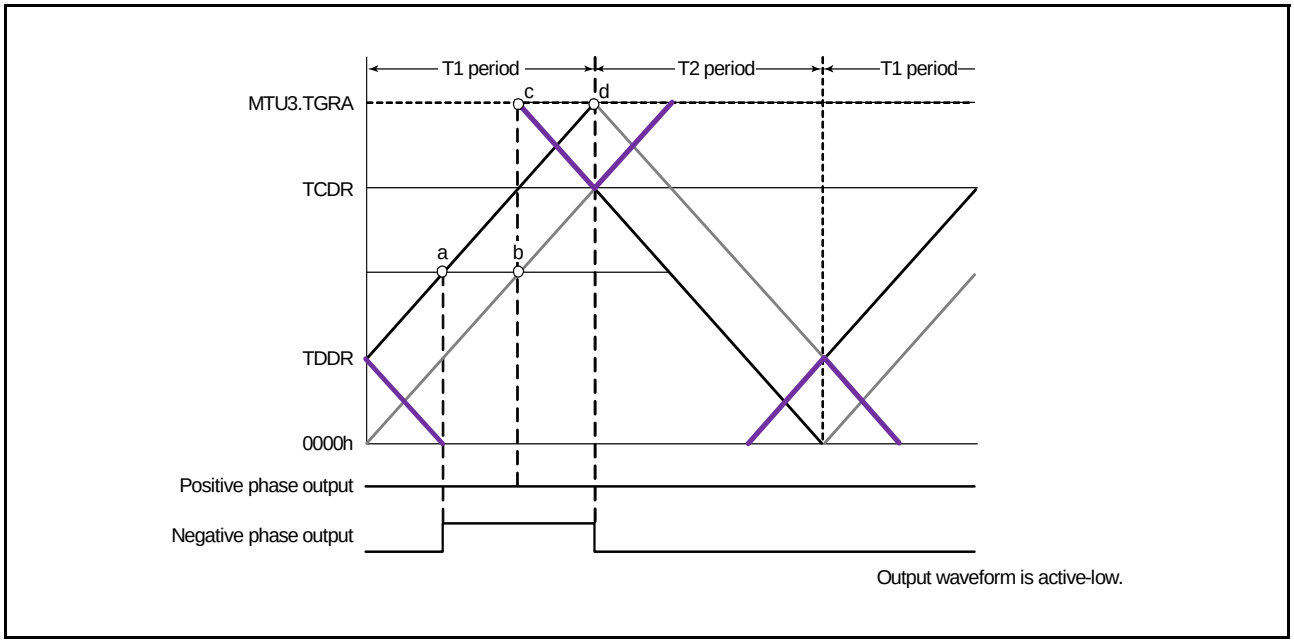



**Figure 20.51 Example of 0% and 100% Waveform Output in Complementary PWM Mode (3)**

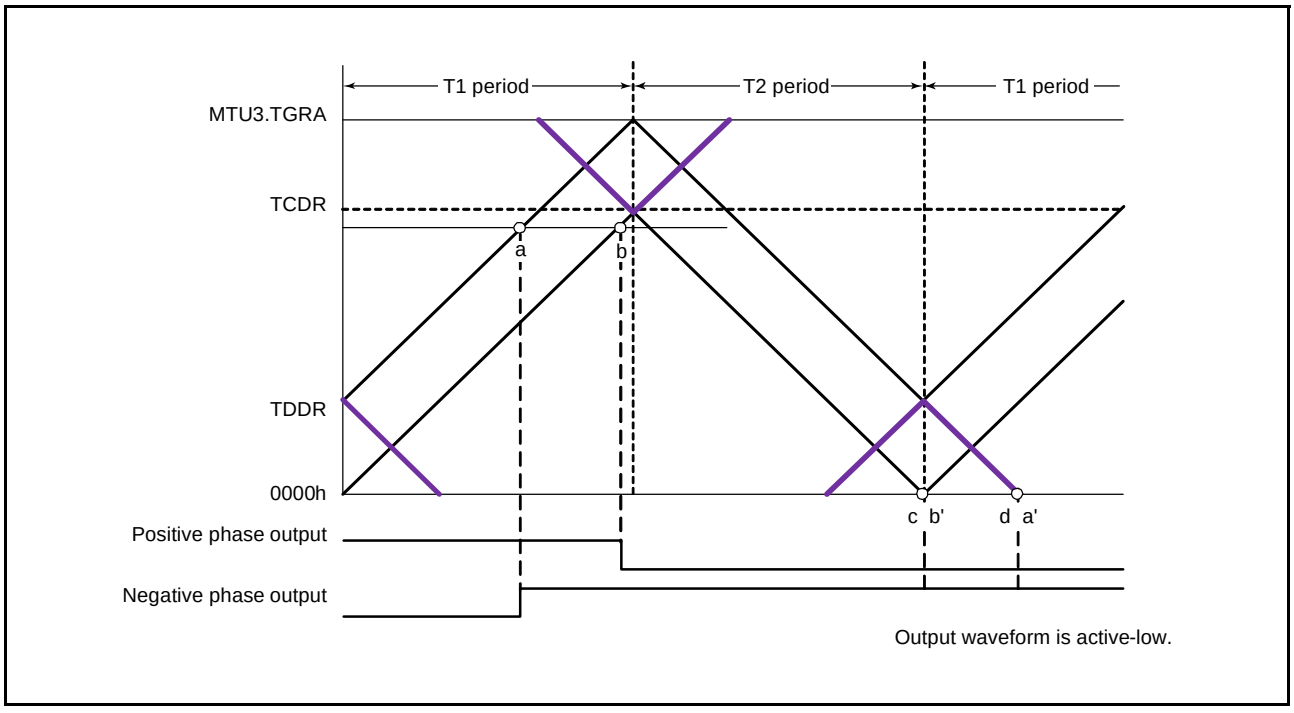



**Figure 20.52 Example of 0% and 100% Waveform Output in Complementary PWM Mode (4)**

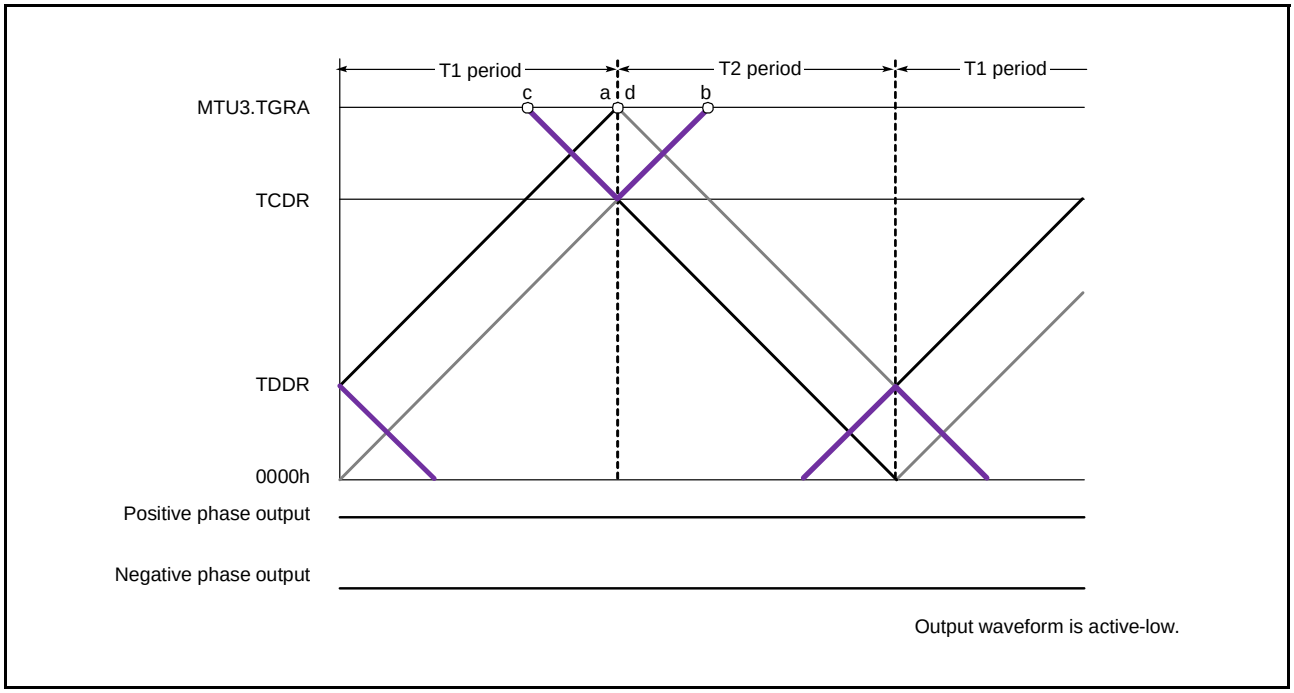



**Figure 20.53 Example of 0% and 100% Waveform Output in Complementary PWM Mode (5)**

### (l) Toggle Output Synchronized with PWM Cycle

In complementary PWM mode, toggle output in synchronization with the PWM carrier cycle can be generated by setting the PSYE bit to 1 in the timer output control register (TOCR). An example of a toggle output waveform is shown in Figure 20.54.

This output is toggled by a compare match between MTU3.TCNT and MTU3.TGRA and a compare match between MTU4.TCNT and 0000h.

The MTIOC3A pin is assigned for this toggle output. The initial output is a high level.

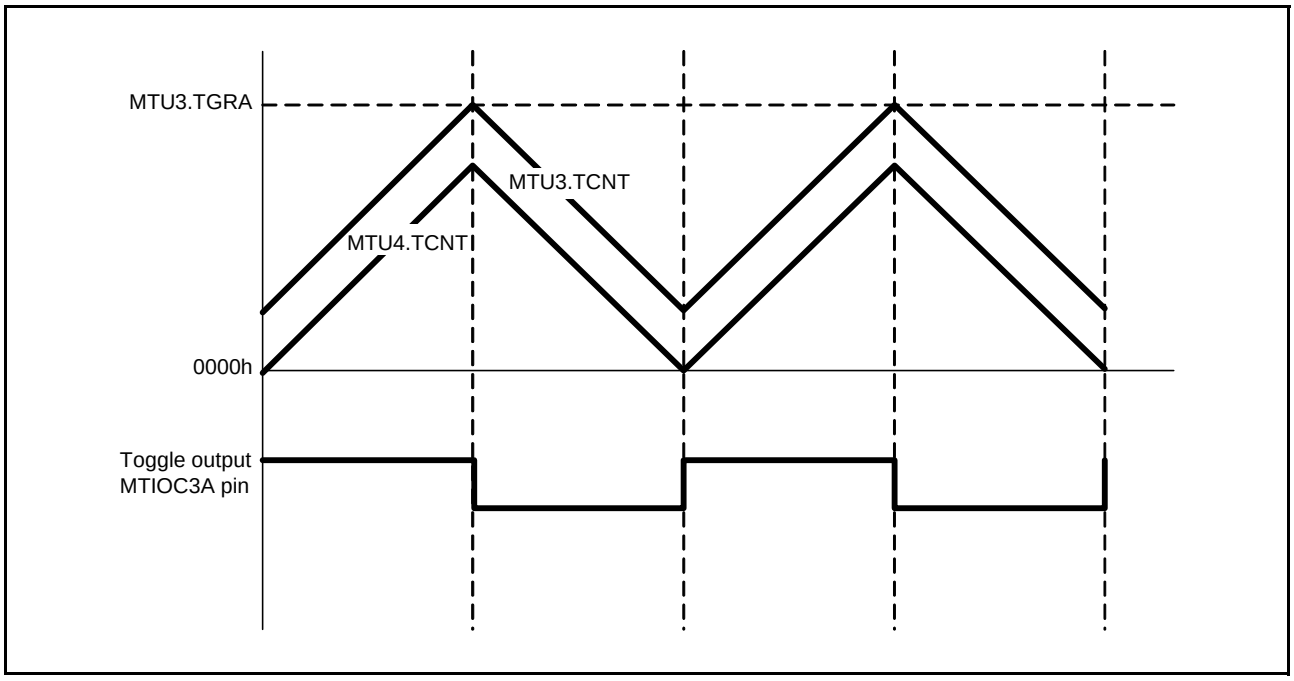



**Figure 20.54 Example of Toggle Output Waveform Synchronized with PWM Output**

### (m) Counter Clearing by Another Channel

In complementary PWM mode, MTU3.TCNT, MTU4.TCNT, and TCNTS can be cleared by another channel when a mode for synchronization with another channel is specified through the timer synchronous register (TSYR) and synchronous clearing is selected with the CCLR[2:0] bits in the timer control register (TCR).

Figure 20.55 illustrates an example of this operation.

Use of this function enables a counter to be cleared and restarted through an external signal.

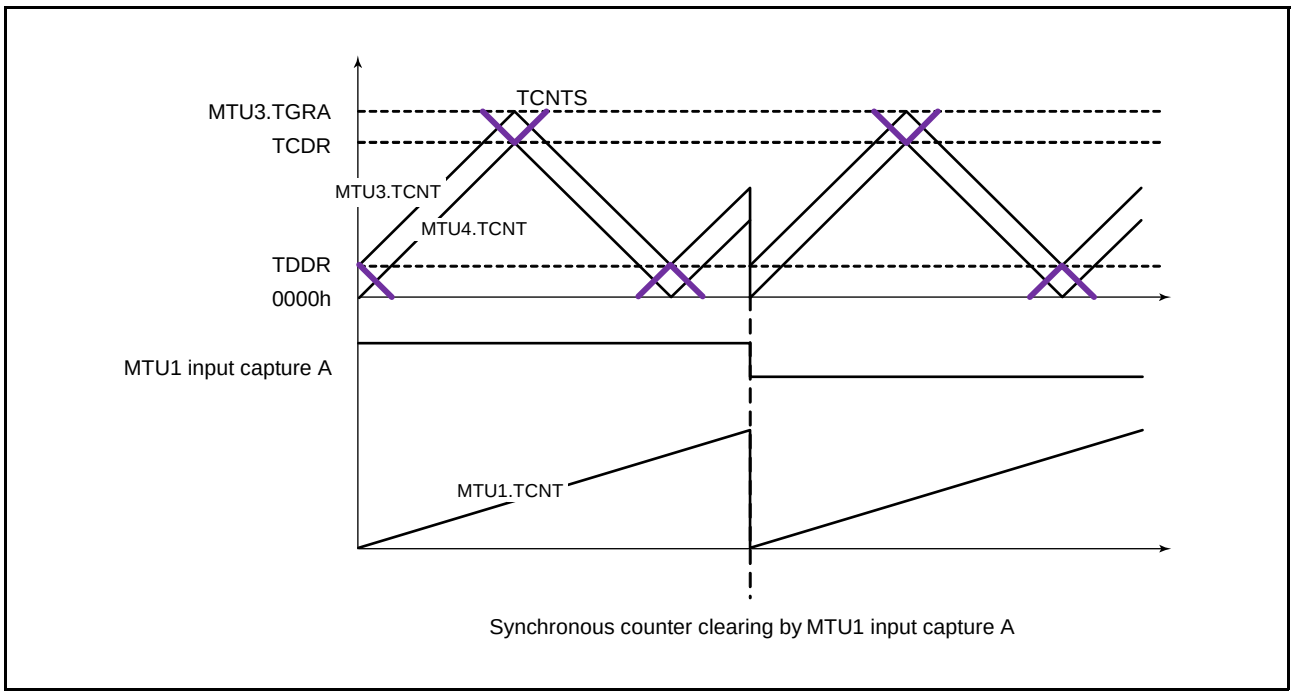



**Figure 20.55 Counter Clearing Synchronized with Another Channel**

### (n) Output Waveform Control at Synchronous Counter Clearing in Complementary PWM Mode

Setting the TWCR.WRE bit to 1 suppresses initial output when synchronous counter clearing occurs in the Tb interval at the trough in complementary PWM mode and controls abrupt change in duty cycle at synchronous counter clearing. Initial output suppression through setting TWCR.WRE bit to 1 is applicable only when synchronous clearing occurs in the Tb interval at the trough as indicated by (10) or (11) in Figure 20.56. When synchronous clearing occurs outside that interval, the initial value specified by the TOCR1.OLSN bit and TOCR1.OLSP bit is output. Even in the Tb interval at the trough, if synchronous clearing occurs in the initial output period (indicated by (1) in Figure 20.56) immediately after the counters start operation, initial value output is not suppressed.
Synchronous clearing generated in MTU0 to MTU2 can cause counter clearing in the MTU.

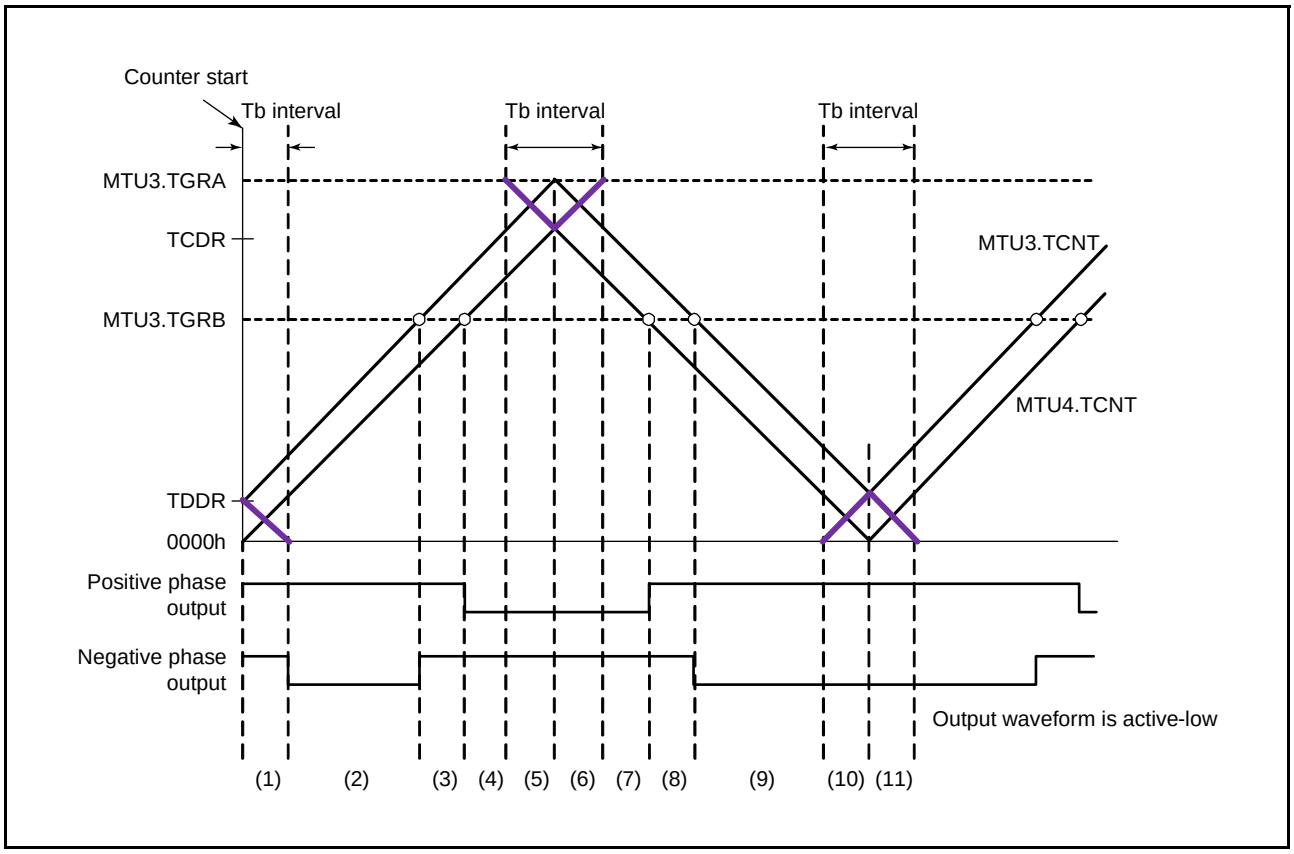



**Figure 20.56 Timing for Synchronous Counter Clearing**

- Example of Procedure for Setting Output Waveform Control at Synchronous Counter Clearing in Complementary PWM Mode

An example of the procedure for setting output waveform control at synchronous counter clearing in complementary PWM mode is shown in Figure 20.57.

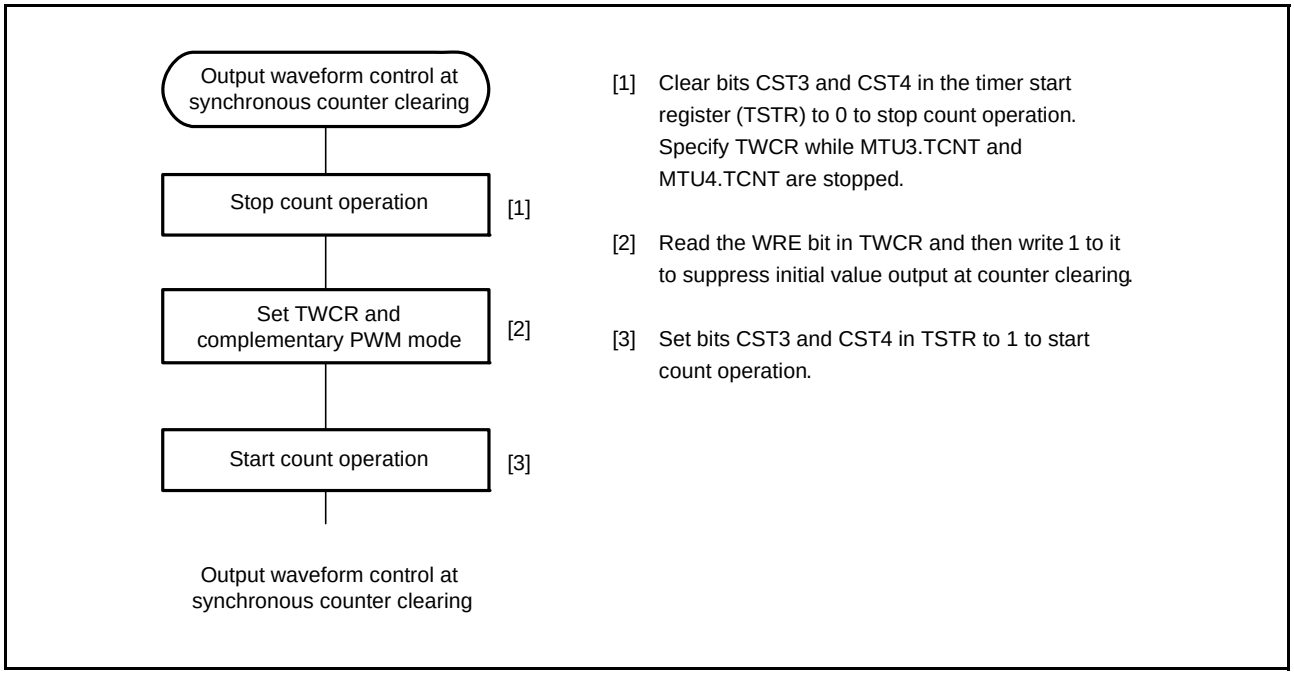



**Figure 20.57 Example of Procedure for Setting Output Waveform Control at Synchronous Counter Clearing in Complementary PWM Mode**

- Examples of Output Waveform Control at Synchronous Counter Clearing in Complementary PWM Mode

Figure 20.58 to Figure 20.61 show examples of output waveform control in which the MTU operates in complementary PWM mode and synchronous counter clearing is generated while the TWCR.WRE bit is set to 1. In the examples shown in Figure 20.58 to Figure 20.61, synchronous counter clearing occurs at timing (3), (6), (8), and (11) shown in Figure 20.56, respectively.

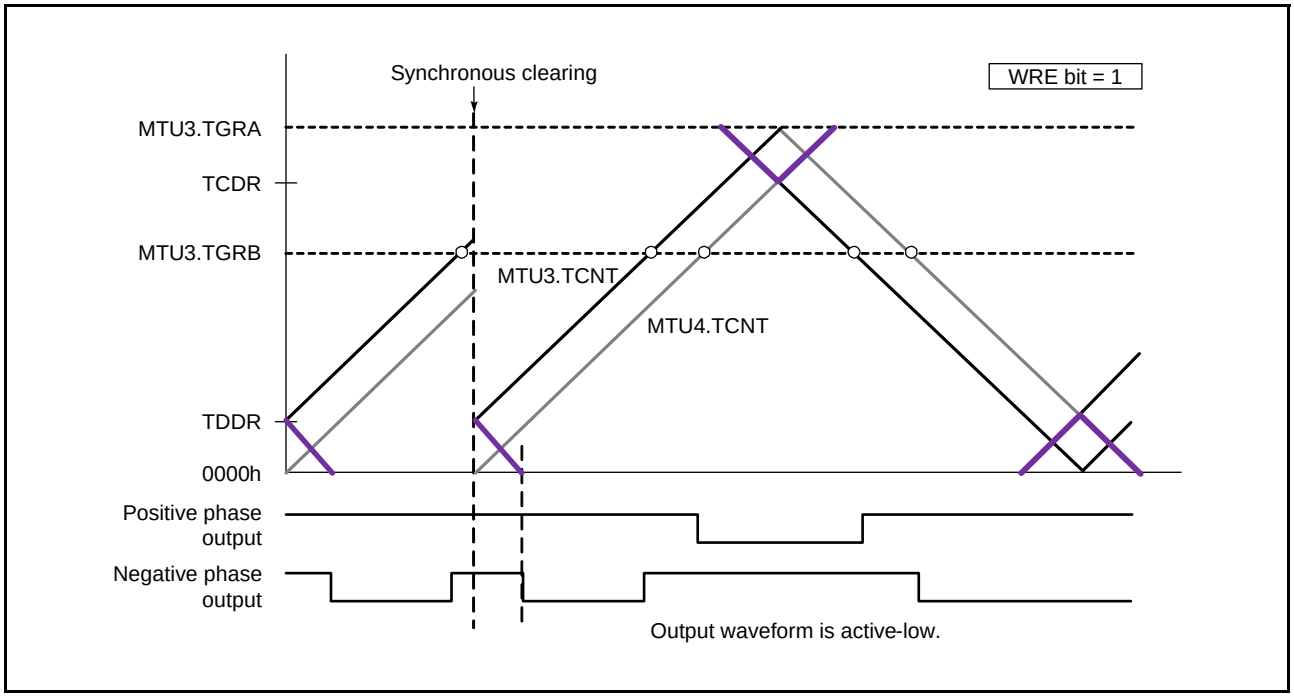



**Figure 20.58 Example of Synchronous Clearing in Dead Time during Up-Counting (Timing (3) in Figure 20.56; TWCR.WRE Bit is 1)**

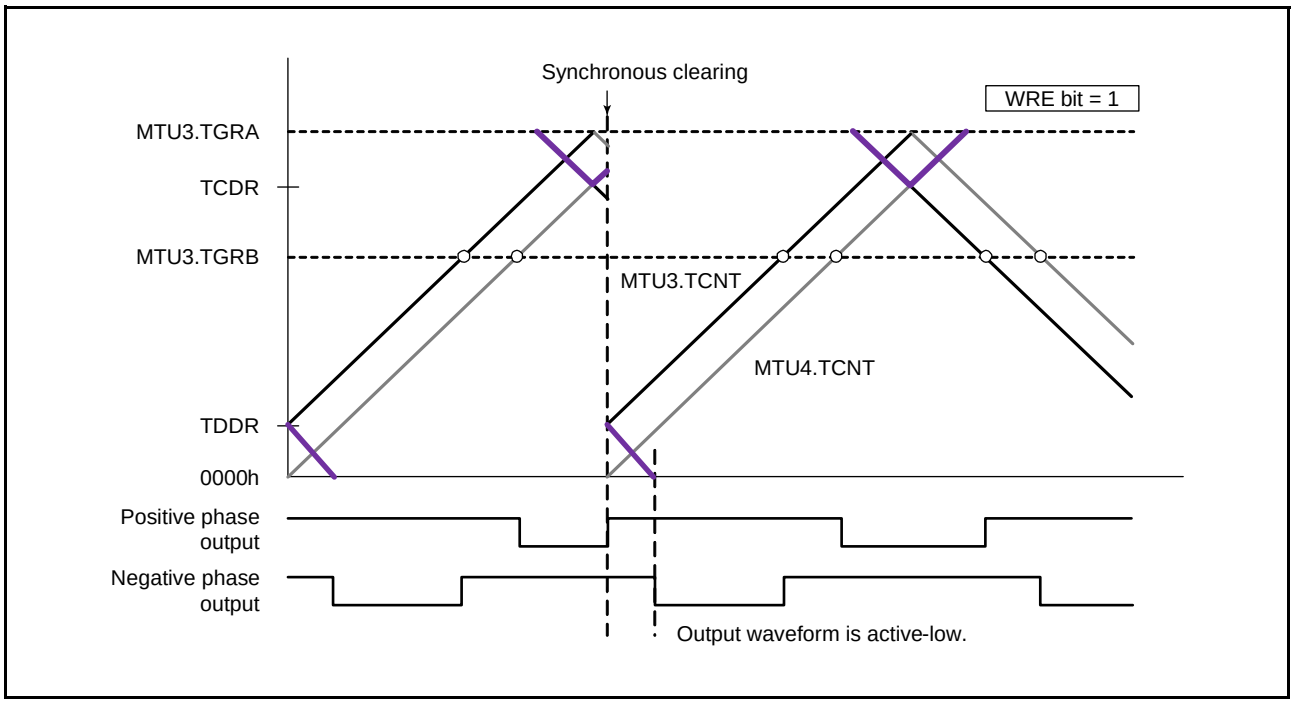



**Figure 20.59 Example of Synchronous Clearing in Interval Tb at Crest (Timing (6) in Figure 20.56; TWCR.WRE Bit is 1)**

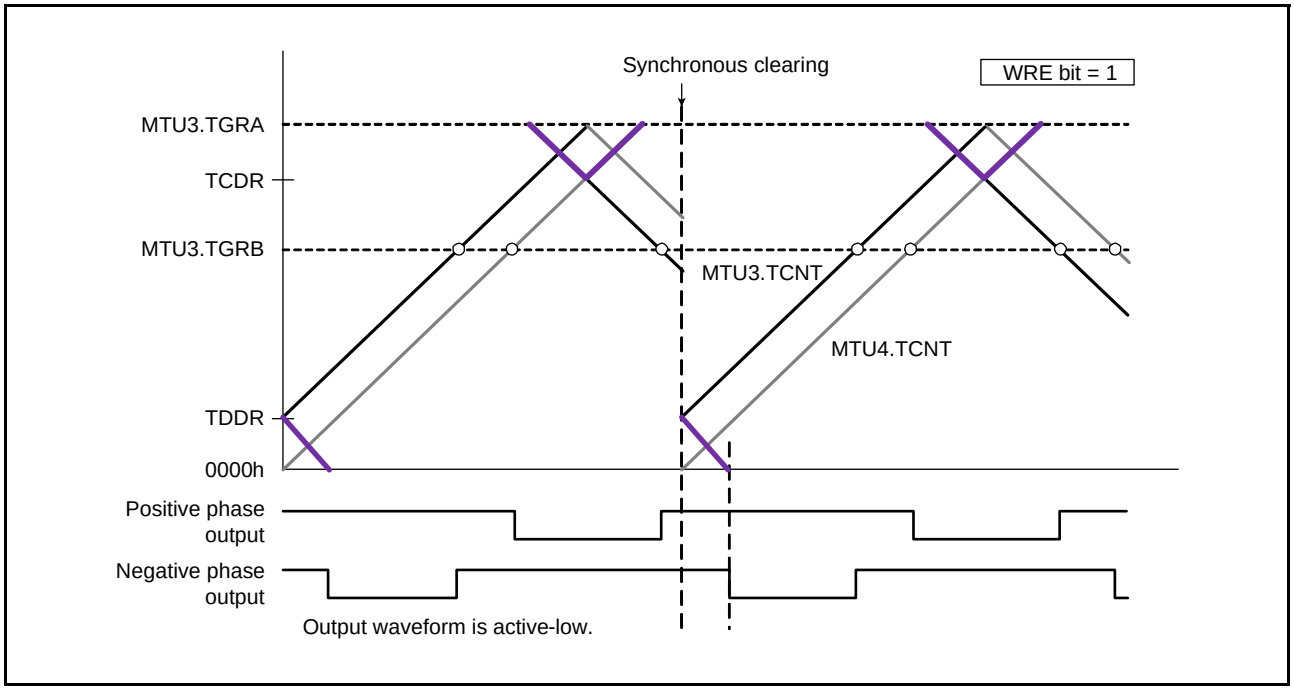



**Figure 20.60 Example of Synchronous Clearing in Dead Time during Down-Counting (Timing (8) in Figure 20.56; TWCR.WRE Bit is 1)**

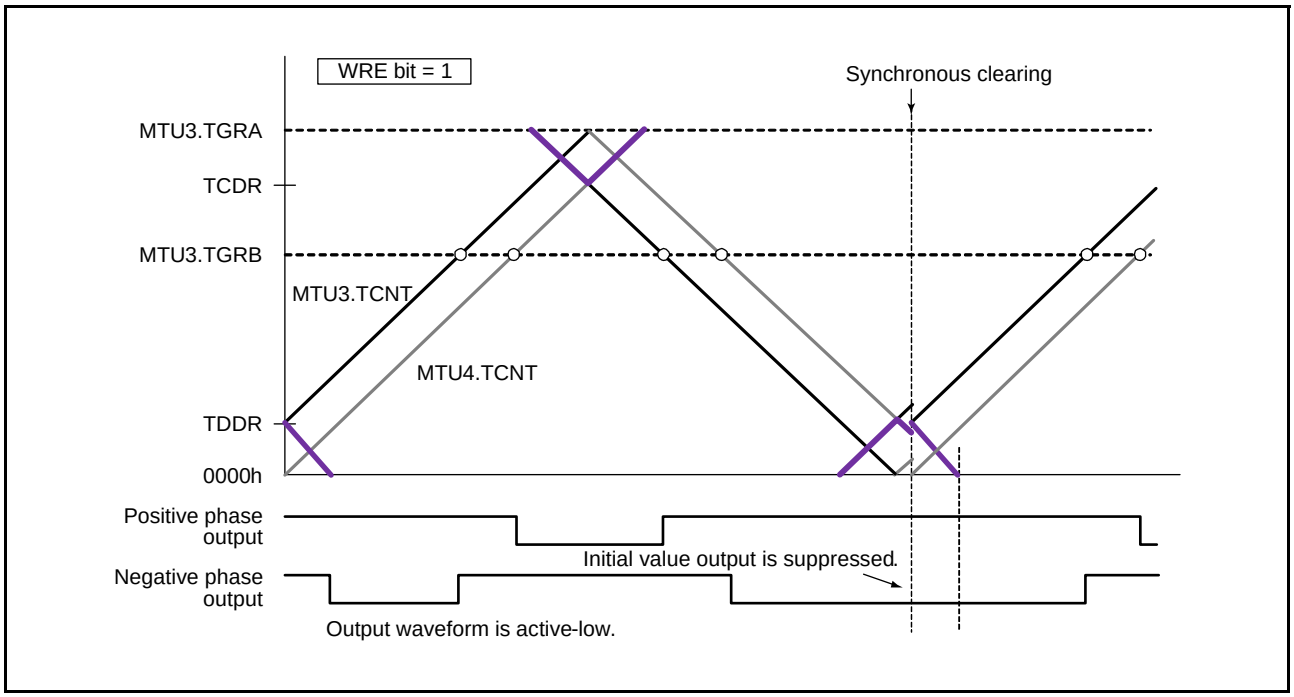



**Figure 20.61 Example of Synchronous Clearing in Interval Tb at Trough (Timing (11) in Figure 20.56; TWCR.WRE Bit is 1)**

### (o) Counter Clearing by MTU3.TGRA Compare Match

In complementary PWM mode, MTU3.TCNT, MTU4.TCNT, and TCNTS can be cleared by MTU3.TGRA compare match when the CCE bit is set in the timer waveform control register (TWCR).

Figure 20.62 shows an operation example.

Note: • Use this function only in complementary PWM mode 1 (transfer at crest).

Note: • Do not specify synchronous clearing by another channel (do not set the SYNC0 to SYNC4 bits in the timer synchronous register (TSYR) to 1).

Note: • Do not set the PWM duty cycle value to 0000h.

Note: • Do not set the PSYE bit in timer output control register 1 (TOCR1) to 1.

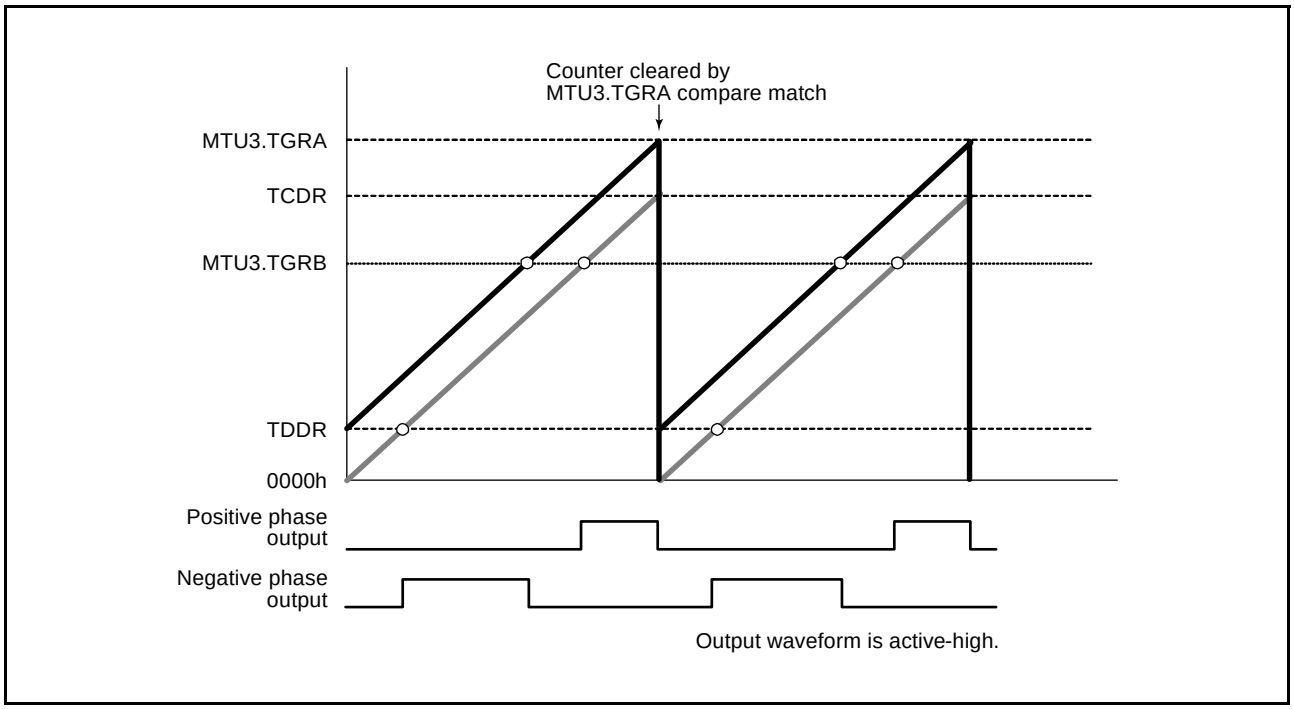



**Figure 20.62 Example of Counter Clearing Operation by MTU3.TGRA Compare Match**

### (p) Example of Waveform Output for Driving AC Synchronous Motor (Brushless DC Motor)

In complementary PWM mode, a brushless DC motor can easily be controlled using the timer gate control register (TGCR). Figure 20.63 to Figure 20.66 show examples of brushless DC motor driving waveforms created using TGCR.
To switch the output phases for a 3-phase brushless DC motor by means of external signals detected with a Hall element, etc., clear the TGCR.FB bit to 0. In this case, the external signals indicating the magnetic pole position should be input to timer input pins MTIOC0A, MTIOC0B, and MTIOC0C in MTU0. When an edge is detected at pin MTIOC0A, MTIOC0B, or MTIOC0C, the output on/off state is switched automatically.
When the TGCR.FB bit is 1, the output on/off state is switched when the TGCR.UF bit, TGCR.VF bit, or TGCR.WF bit is cleared to 0 or set to 1.
The driving waveforms are output from the 6-phase output pins for complementary PWM mode.
With this 6-phase output, while the output is turned on, chopping output is available through complementary PWM mode output function by setting the TGCR.N bit or TGCR.P bit to 1. When the TGCR.N bit or TGCR.P bit is 0, the level output is selected.
The active level of the 6-phase output (on output level) can be set with the TOCR1.OLSN bit and TOCR1.OLSP bit regardless of the setting of the TGCR.N bit and TGCR.P bit.

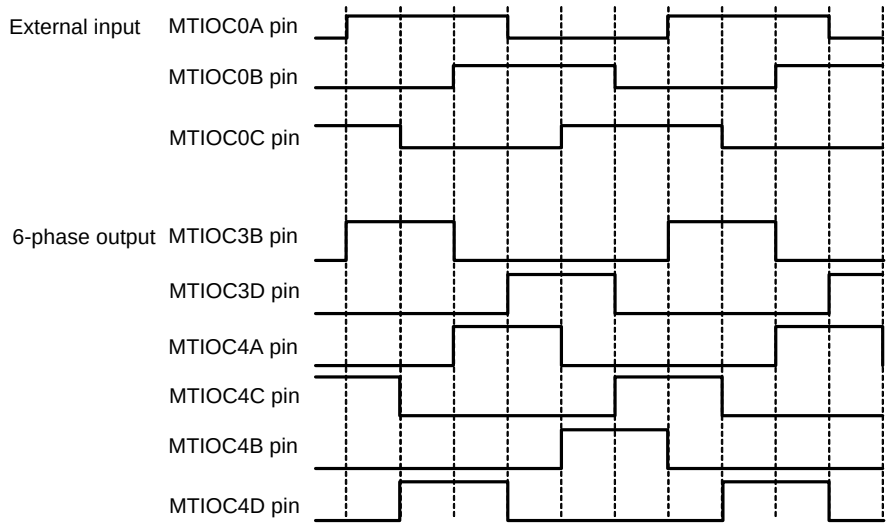



**Figure 20.63 Example of Output Phase Switching by External Input (1)**

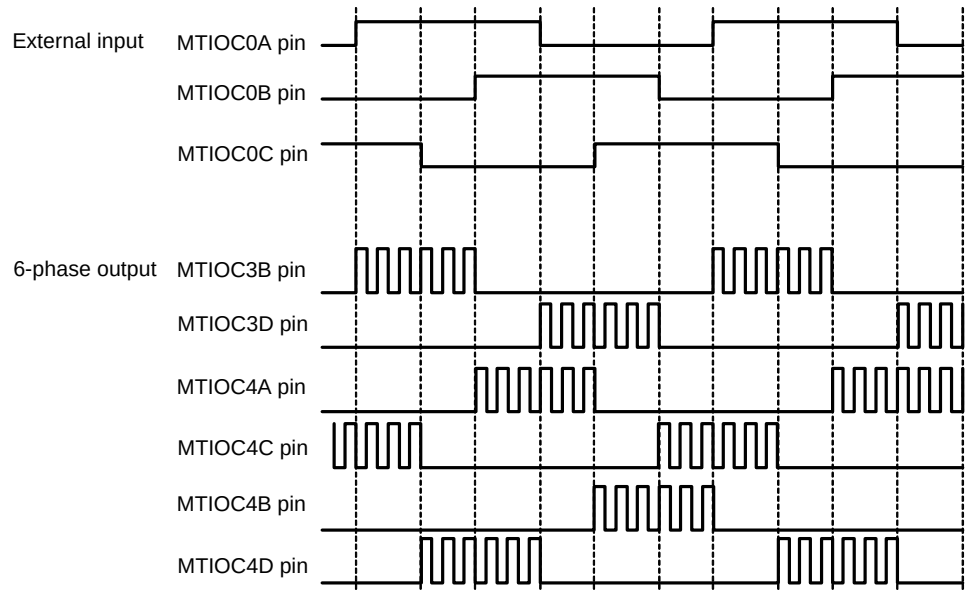



**Figure 20.64 Example of Output Phase Switching by External Input (2)**

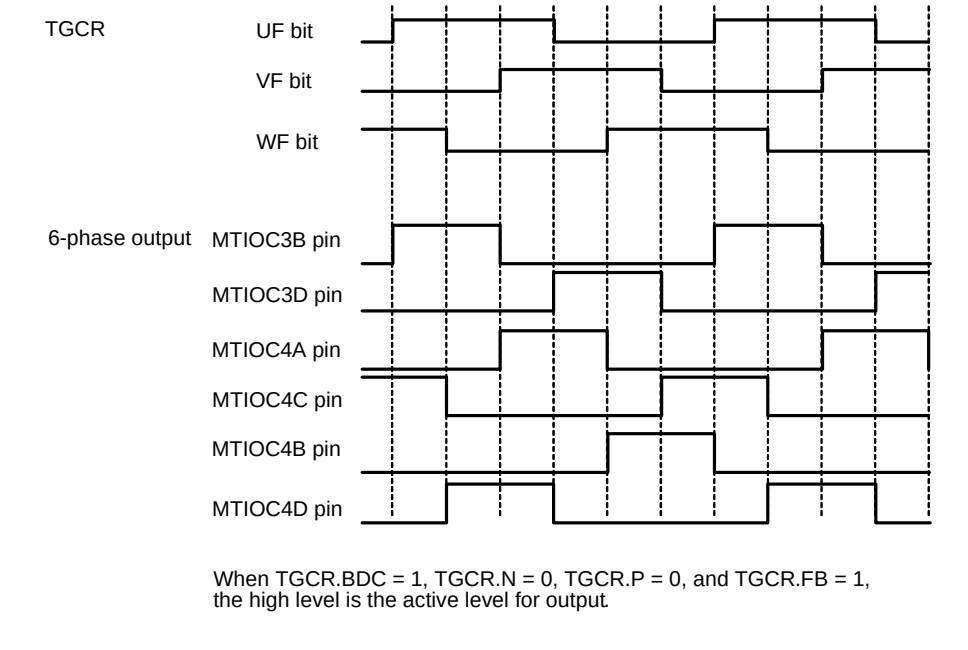



**Figure 20.65 Example of Output Phase Switching through UF, VF, and WF Bit Settings (1)**

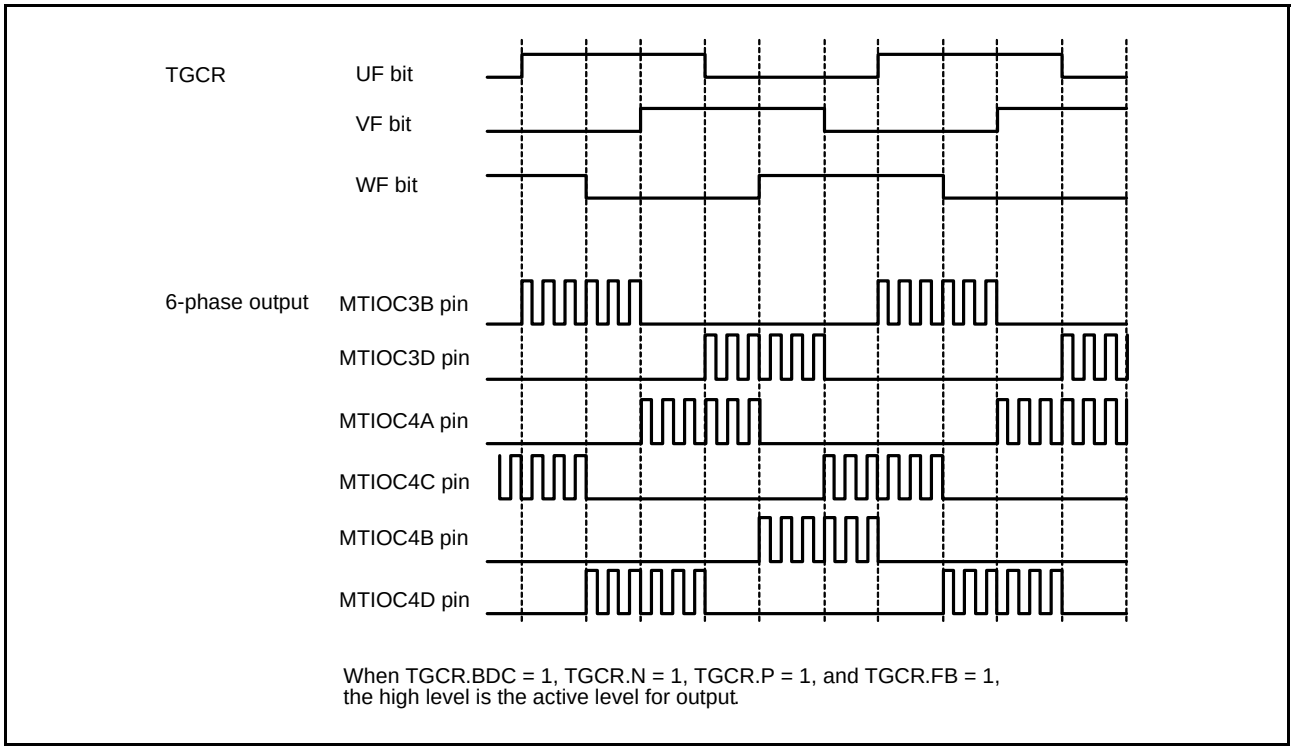



**Figure 20.66 Example of Output Phase Switching through UF, VF, and WF Bit Settings (2)**

### (q) A/D Converter Start Request Setting

In complementary PWM mode, an A/D converter start request can be issued using MTU3.TGRA compare match, MTU4.TCNT underflow (trough), or compare match on a channel other than MTU3 and MTU4.

When start requests using MTU3.TGRA compare match are specified, A/D conversion can be started at the crest of the MTU3.TCNT count.

A/D converter start requests can be specified by setting the TIER.TTGE bit to 1. To issue an A/D converter start request at an MTU4.TCNT underflow (trough), set the MTU4.TIER.TTGE2 bit to 1.

### (3) Interrupt Skipping in Complementary PWM Mode

Interrupts TGIA3 (at the crest) and TCIV4 (at the trough) in MTU3 and MTU4 can be skipped up to seven times by setting the timer interrupt skipping set register (TITCR).
Transfers from a buffer register to a temporary register or a compare register can be skipped in coordination with interrupt skipping by making settings in the timer buffer transfer set register (TBTER). For the linkage with buffer registers, refer to description (c), Buffer Transfer Control Linked with Interrupt Skipping, below.
A/D converter start requests generated by the A/D converter start request delaying function can also be skipped in coordination with interrupt skipping by making settings in the timer A/D converter request control register (TADCR). For the linkage with the A/D converter start request delaying function, refer to section 20.3.9, A/D Converter Start Request Delaying Function.
The timer interrupt skipping set register (TITCR) should be set while the TGIA3 and TCIV4 interrupt requests are disabled by the settings of MTU3.TIER and MTU4.TIER under the conditions in which compare match never occur and TGIA3 and TGIA4 interrupt requests by compare match are never generated. Before changing the skipping count, be sure to clear the TITCR.T3AEN and TITCR.T4VEN bits to 0 to clear the skipping counter.

#### (a) Example of Interrupt Skipping Operation Setting Procedure

Figure 20.67 shows an example of the interrupt skipping operation setting procedure. Figure 20.68 shows the periods during which interrupt skipping count can be changed.

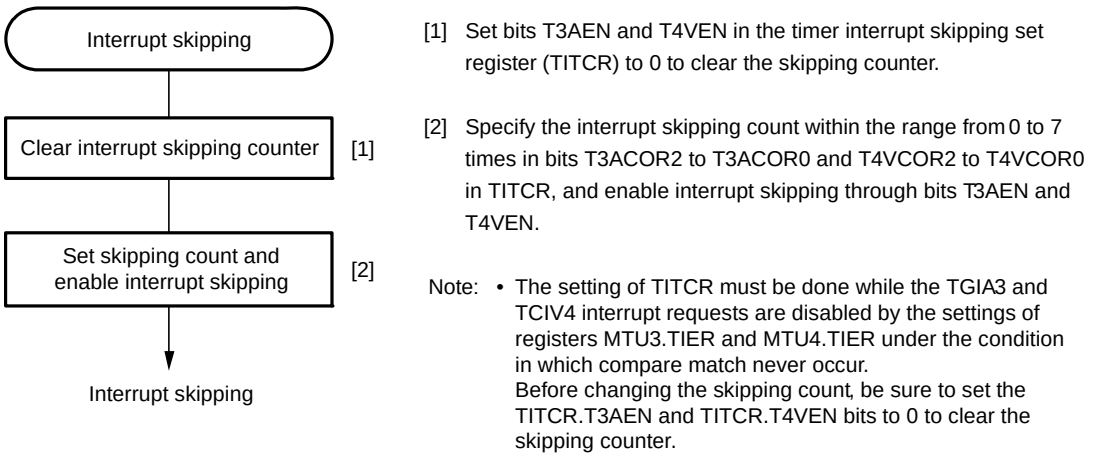



**Figure 20.67 Example of Interrupt Skipping Operation Setting Procedure**

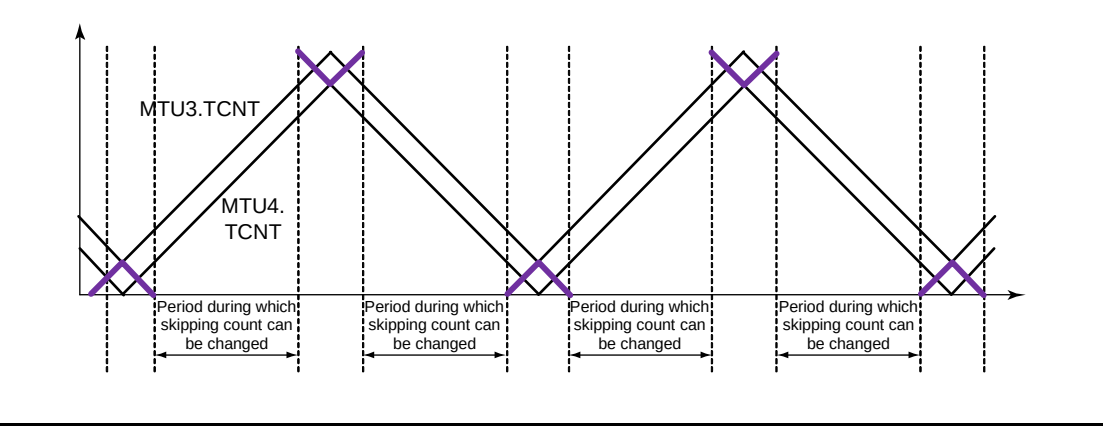



**Figure 20.68 Periods during which Interrupt Skipping Count can be Changed**

### (b) Example of Interrupt Skipping Operation

Figure 20.69 shows an example of MTU3.TGIA interrupt skipping in which the interrupt skipping count is set to three by the TITCR.T3ACOR bit and the TITCR.T3AEN bit is set to 1.

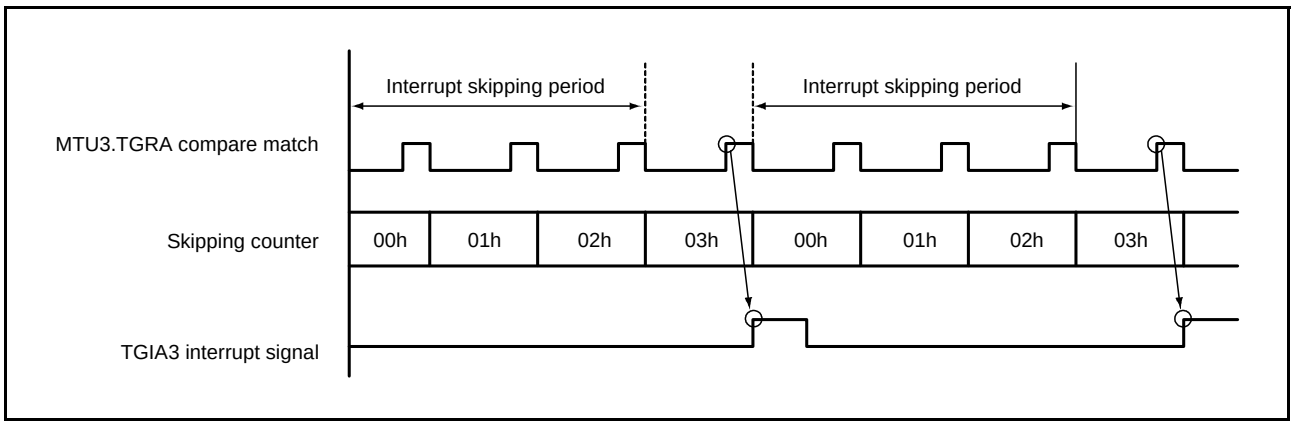



**Figure 20.69 Example of Interrupt Skipping Operation**

### (c) Buffer Transfer Control Linked with Interrupt Skipping

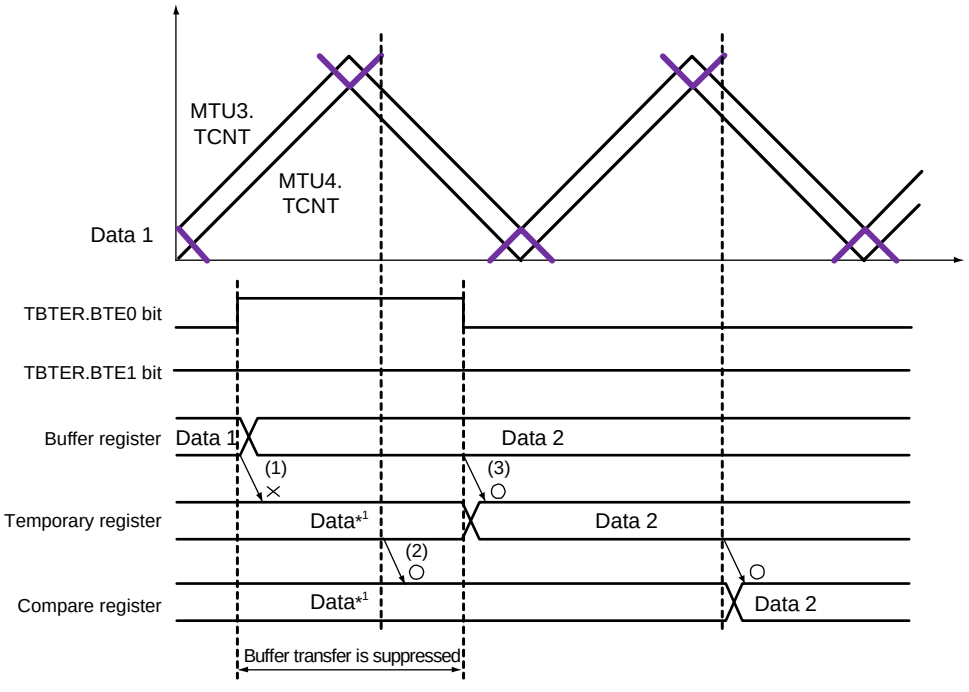
In complementary PWM mode, whether to transfer data from a buffer register to a temporary register and whether to link the transfer with interrupt skipping can be specified with the BTE[1:0] bits in the timer buffer transfer set register (TBTER).

Figure 20.70 shows an example of operation when buffer transfer is disabled (BTE[1:0] = 01b). While this setting is valid, data is not transferred from the buffer register to the temporary register.

Figure 20.71 shows an example of operation when buffer transfer is linked with interrupt skipping (BTE[1:0] = 10b). While this setting is valid, if data is written to the buffer register within the buffer transfer-enabled period, the data is transferred immediately from the buffer register to the temporary register. If data is written to the buffer register outside the buffer transfer-enabled period, the data is transferred from the buffer register to the temporary register at the timing when the next buffer transfer-enabled period starts.

Note that the buffer transfer-enabled period depends on the TITCR.T3AEN bit and TITCR.T4VEN bit settings. Figure 20.72 shows the relationship between the TITCR.T3AEN bit and TITCR.T4VEN bit settings and buffer transfer-enabled period.

Note: • This function must always be used in combination with interrupt skipping.
When interrupt skipping is disabled (the T3AEN and T4VEN bits in the timer interrupt skipping set register (TITCR) are cleared to 0 or the skipping count setting bits (T3ACOR and T4VCOR) in TITCR are cleared to 0), make sure that buffer transfer is not linked with interrupt skipping (clear the BTE1 bit in the timer buffer transfer set register (TBTER) to 0).
If buffer transfer is linked with interrupt skipping while interrupt skipping is disabled, buffer transfer is never performed.




(1) No data is transferred from the buffer register to the temporary register in the buffer transfer -disabled period (setting TBTER.BTE[1:0] to 01b).
(2) Data is transferred from the temporary register to the compare register even in the compare transfer-disabled period.
(3) After buffer transfer is enabled, data is transferred from the buffer register to the temporary register.

Note 1. When buffer transfer at the crest is selected.

**Figure 20.70 Example of Operation When Buffer Transfer is Disabled (TBTER.BTE[1:0] = 01b)**

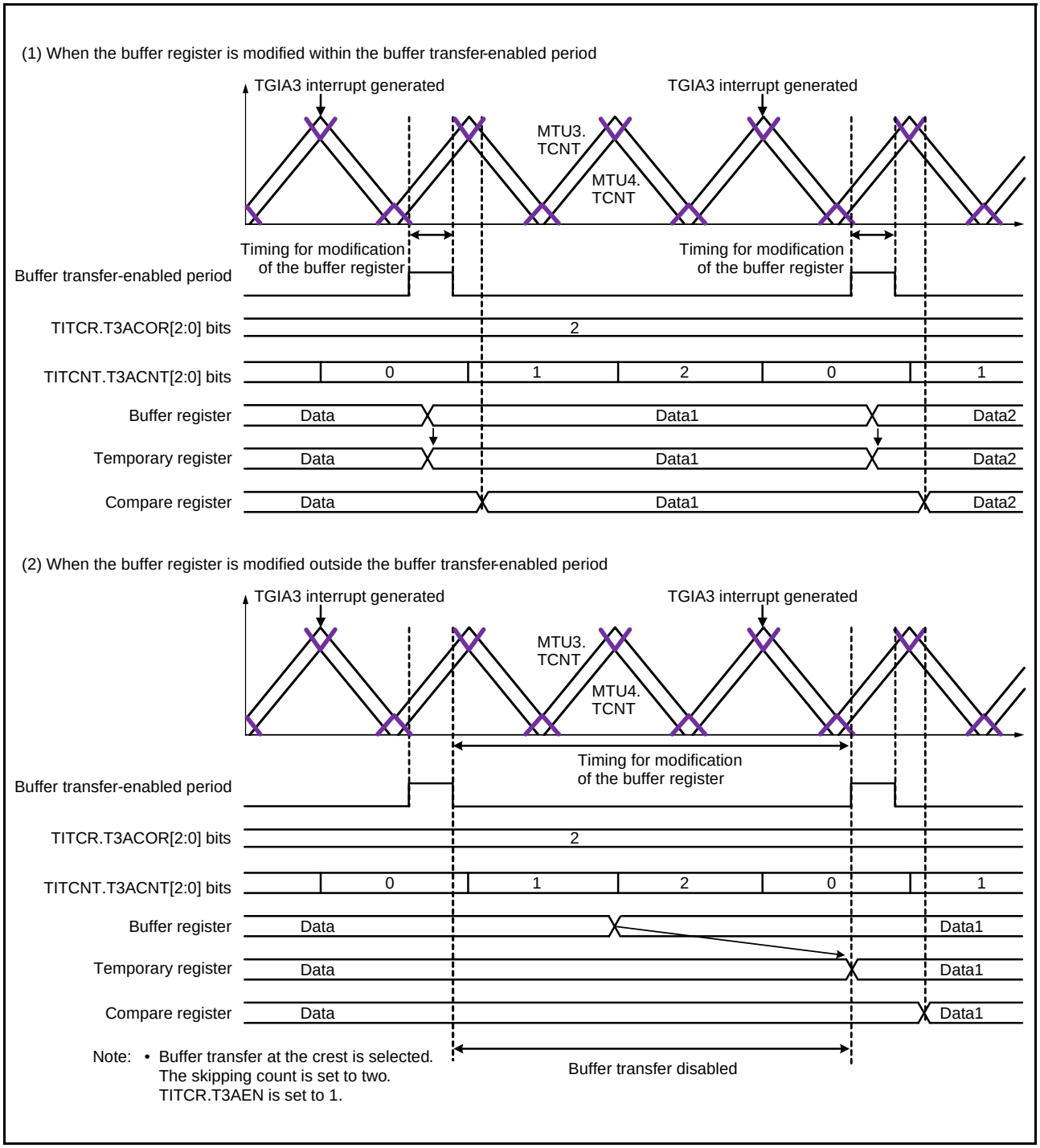



**Figure 20.71 Example of Operation When Buffer Transfer is Linked with Interrupt Skipping (TBTER.BTE[1:0] = 10b)**

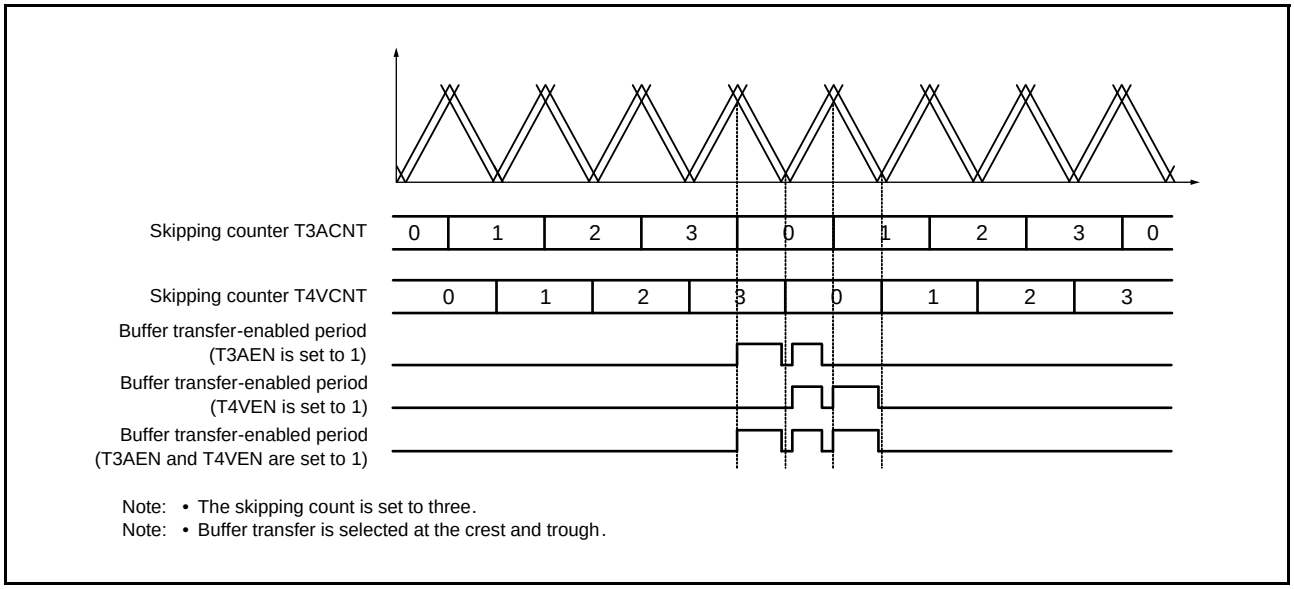



**Figure 20.72 Relationship between Bits T3AEN and T4VEN in TITCR and Buffer Transfer-Enabled Period**

### (4) Complementary PWM Mode Output Protection Functions

The MTU provides the following protection functions for complementary PWM mode output.

#### (a) Register and Counter Miswrite Prevention Function

With the exception of the buffer registers, which can be modified at any time, access by the CPU can be enabled or disabled for the mode registers, control registers, compare registers, and counters used in complementary PWM mode by means of the TRWER.RWE bit. The applicable registers are some of the registers in MTU3 and MTU4 shown below:

22 registers in total

MTU3.TCR and MTU4.TCR, MTU3.TMDR and MTU4.TMDR, MTU3.TIORH and MTU4.TIORH, MTU3.TIORL and MTU4.TIORL, MTU3.TIER and MTU4.TIER, MTU3.TCNT and MTU4.TCNT, MTU3.TGRA and MTU4.TGRA, MTU3.TGRB and MTU4.TGRB, TOER, TOCR1, TOCR2, TGCR, TCDR, and TDDR

This function can disable CPU access to the mode registers, control registers, and counters to prevent miswriting due to CPU runaway. In the access-disabled state, the applicable registers are read as undefined and writing to these registers is ignored.

#### (b) Halting of PWM Output by External Signal

The 6-phase PWM output pins can be set to the high-impedance state automatically by inputting specified external signals.
Refer to section 21, Port Output Enable 2 (POE2a), for details.

#### (c) Halting of PWM Output when Oscillator is Stopped

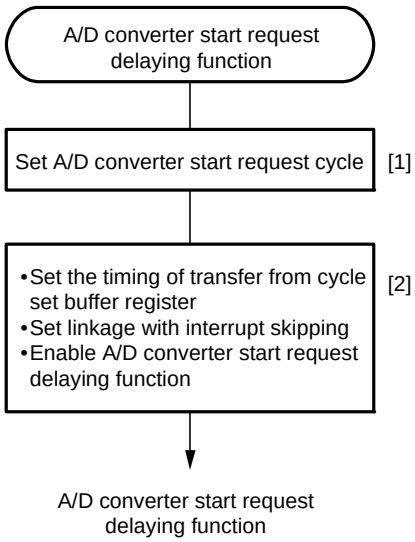
Upon detecting that the clock input to this MCU has stopped, the 6-phase PWM output pins are automatically set to the high-impedance state. Note that the pin states are not guaranteed when the clock is restarted.
Refer to section 9.5, Oscillation Stop Detection Function.

### 20.3.9 A/D Converter Start Request Delaying Function

A/D converter start requests can be issued in MTU4 by making settings in the timer A/D converter start request control register (TADCR), timer A/D converter start request cycle set registers (MTU4.TADCORA and MTU4.TADCORB), and timer A/D converter start request cycle set buffer registers (MTU4.TADCOBRA and MTU4.TADCOBRB).
The A/D converter start request delaying function compares MTU4.TCNT with MTU4.TADCORA or MTU4.TADCORB, and when their values match, the function issues a respective A/D converter start request (TRG4AN or TRG4BN).
A/D converter start requests (TRG4AN and TRG4BN) can be skipped in coordination with interrupt skipping by making settings in the TADCR.ITA3AE bit, TADCR.ITA4VE bit, TADCR.ITB3AE bit, and ITB4VE bit.

#### (1) Example of Procedure for Specifying A/D Converter Start Request Delaying Function

Figure 20.73 shows an example of procedure for specifying the A/D converter start request delaying function.




[1] Set the cycle in the timer A/D converter start request cycle set buffer registers A and B (MTU4.TADCOBRA/TADCOBRB) and timer A/D converter start request cycles set registers A and B (MTU4.TADCORA/TADCORB). (The same initial value must be specified in the cycle buffer register and cycle register.)

[2] Use the BF[1:0] bits in the timer A/D converter start request control register (TADCR) to specify the timing of buffer transfer from the timer A/D converter start request cycle buffer register to A/D converter start request cycle register.
- Specify whether to link with interrupt skipping through TADCR.ITA3AE bit, TADCR.ITA4VE bit, TADCR.ITB3AE bit, and TADCR.ITB4VE bit.
- Use TADCR.UT4AE bit, TADCR.DT4AE bit, TADCR.UT4BE bit, and TADCR.DT4BE bit to enable A/D converter start requests (TRG4AN or TRG4BN).

Note: Perform TADCR setting while MTU4.TCNT is stopped. Do not set TADCR.BF1 bit, TADCR.ITA3AE bit, TADCR.ITA4VE bit, TADCR.ITB3AE bit, TADCR.ITB4VE bit, TADCR.DT4AE bit, or TADCR.DT4BE bit to 1 when complementary PWM mode is not selected.

**Figure 20.73 Example of Procedure for Specifying A/D Converter Start Request Delaying Function**

### (2) Basic Example of A/D Converter Start Request Delaying Function Operation

Figure 20.74 shows a basic example of A/D converter start request signal (TRG4AN) operation when the trough of MTU4.TCNT is specified for the buffer transfer timing and an A/D converter start request signal is output during MTU4.TCNT down-counting.

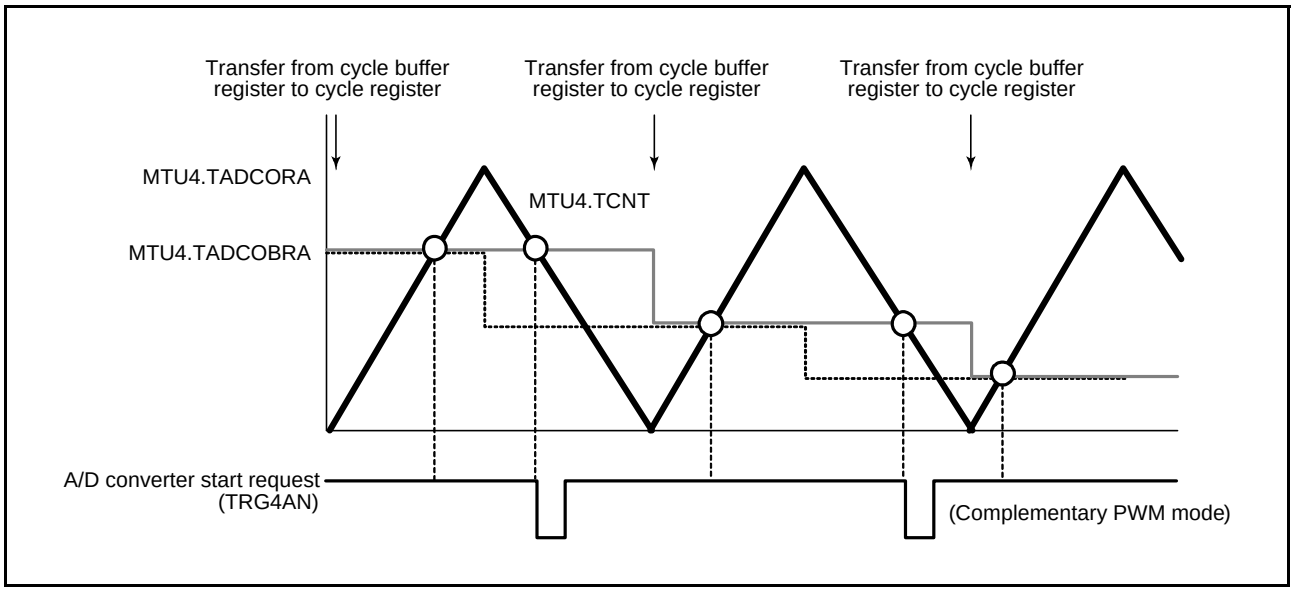



**Figure 20.74 Basic Example of A/D Converter Start Request Signal (TRG4AN) Operation**

### (3) Buffer Transfer

The data in the timer A/D converter start request cycle set registers (MTU4.TADCORA and MTU4.TADCORB) is updated by writing data to the timer A/D converter start request cycle set buffer registers (MTU4.TADCOBRA and MTU4.TADCOBRB). Data is transferred from the buffer registers to the respective cycle set registers at the timing selected with the MTU4.TADCR.BF[1:0] bits.

### (4) A/D Converter Start Request Delaying Function Linked with Interrupt Skipping

A/D converter start requests (TRG4AN and TRG4BN) can be issued in coordination with interrupt skipping by making settings in the TADCR.ITA3AE bit, TADCR.ITA4VE bit, TADCR.ITB3AE bit, and TADCR.ITB4VE bit. Figure 20.75 shows an example of A/D converter start request signal (TRG4AN) operation when TRG4AN output is enabled during MTU4.TCNT up-counting and down-counting and A/D converter start requests are linked with interrupt skipping. Figure 20.76 shows another example of A/D converter start request signal (TRG4AN) operation when TRG4AN output is enabled during MTU4.TCNT up-counting and A/D converter start requests are linked with interrupt skipping.

Note: • This function should be used in combination with interrupt skipping.
When interrupt skipping is disabled (the TITCR.T3AEN bit and TITCR.T4VEN bit are cleared to 0 or the skipping count setting bits (T3ACOR and T4VCOR) in TITCR are cleared to 0), make sure that A/D converter start requests are not linked with interrupt skipping (clear the TADCR.ITA3AE bit, TADCR.ITA4VE bit, TADCR.ITB3AE bit, and TADCR.ITB4VE bit to 0).
Note that TRG4ABN (TRG4AN or TRG4BN) is output as the A/D converter start request signal in this case.

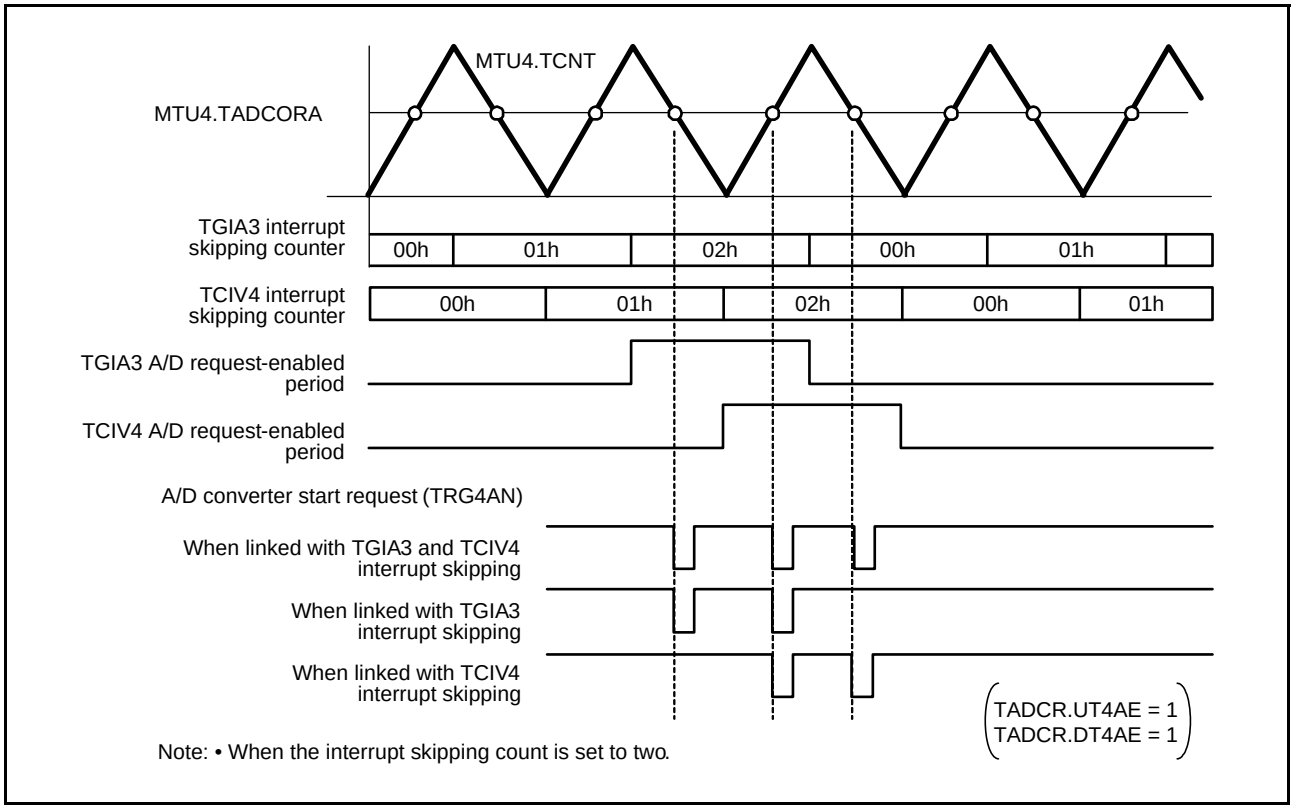



**Figure 20.75 Example of A/D Converter Start Request Signal (TRG4AN) Operation Linked with Interrupt Skipping (when the output of TRG4AN in counting up and down by TCNT is enabled)**

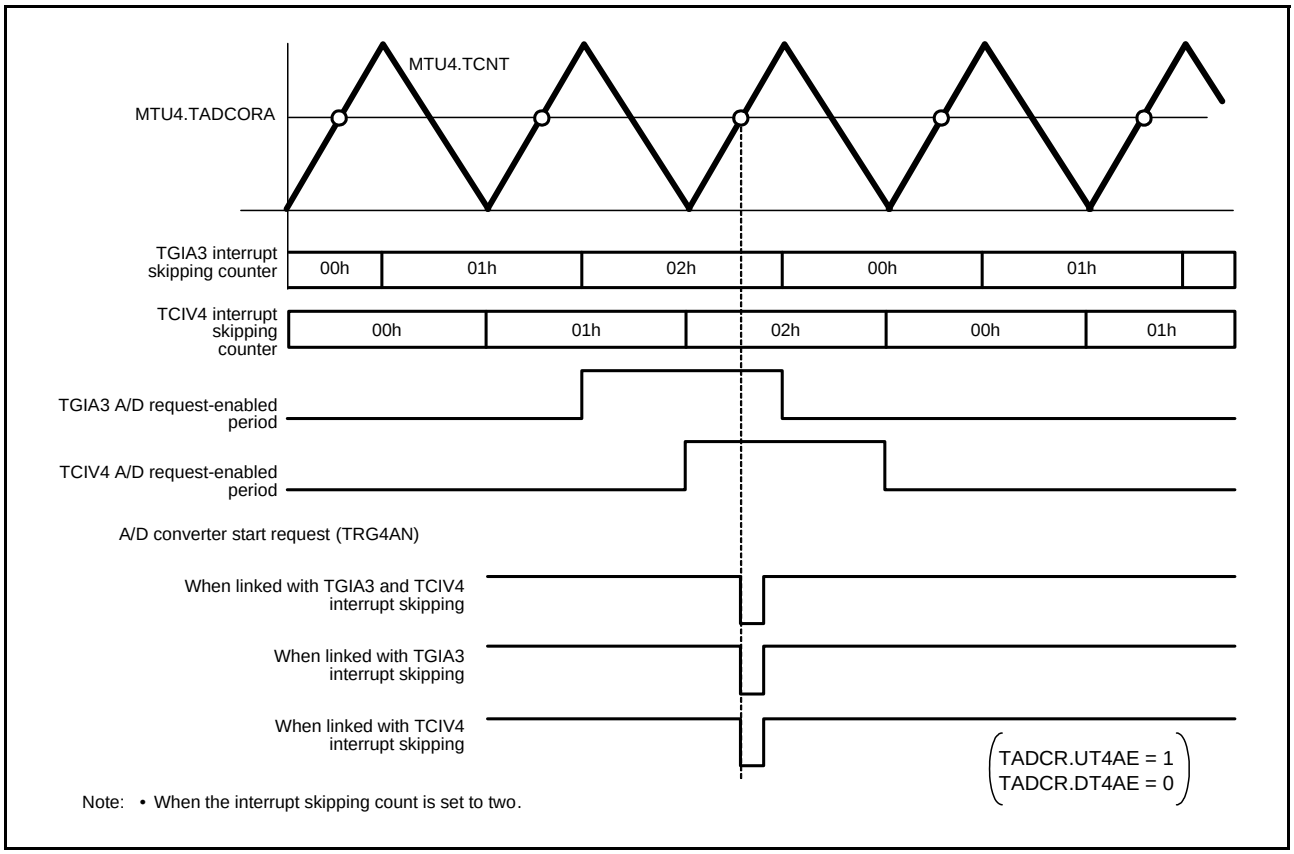



**Figure 20.76 Example of A/D Converter Start Request Signal (TRG4AN) Operation Linked with Interrupt Skipping (when the output of TRG4AN in counting up by TCNT is enabled)**

## 20.3.10 External Pulse Width Measurement

Up to three external pulse widths can be measured in MTU5.

### (1) Example of External Pulse Width Measurement Setting Procedure

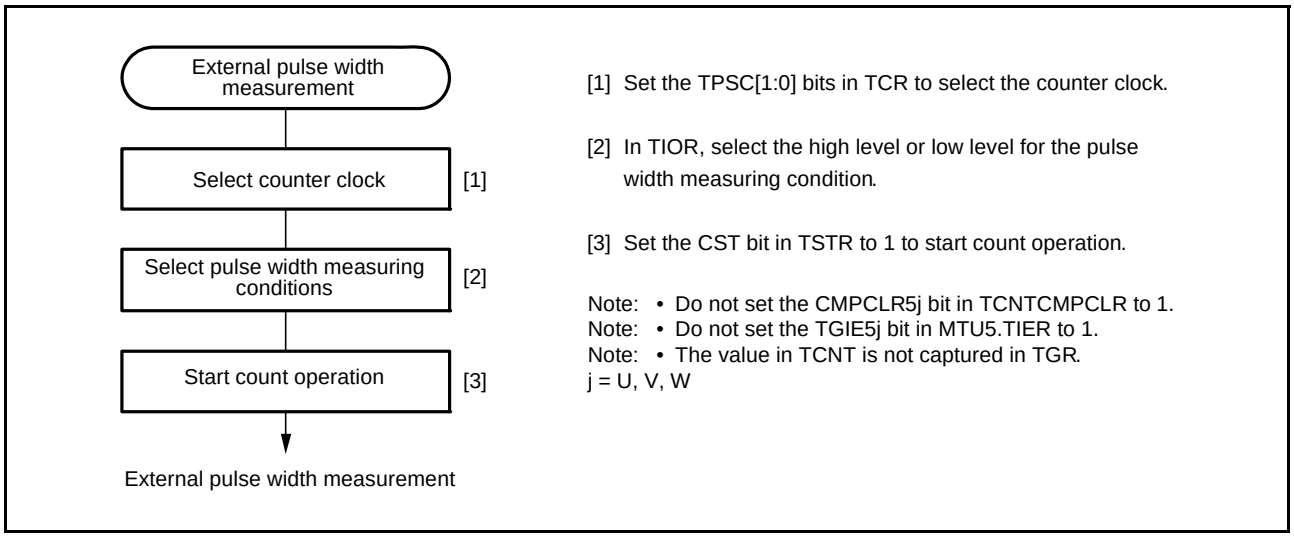



**Figure 20.77 Example of External Pulse Width Measurement Setting Procedure**

### (2) Example of External Pulse Width Measurement

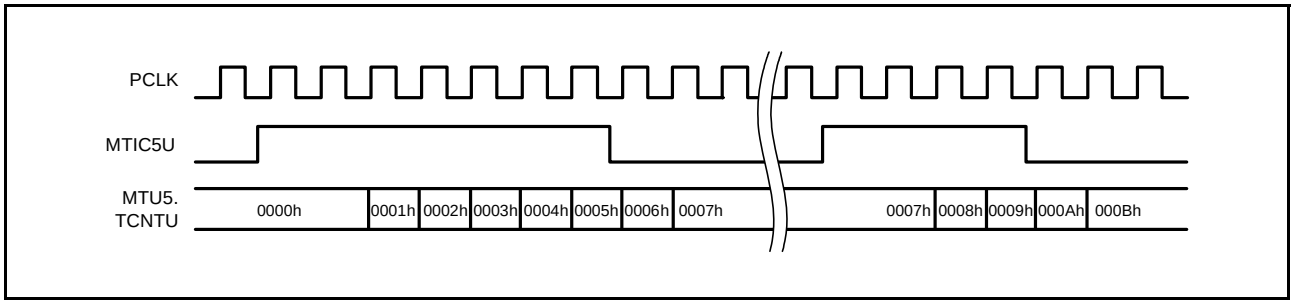



**Figure 20.78 Example of External Pulse Width Measurement (Measuring High Pulse Width)**

### 20.3.11 Dead Time Compensation

By measuring the delay of the output waveform and reflecting it to duty, the external pulse width measurement function can be used as the dead time compensation function to PWM output waveform while the complementary PWM mode is in operation.

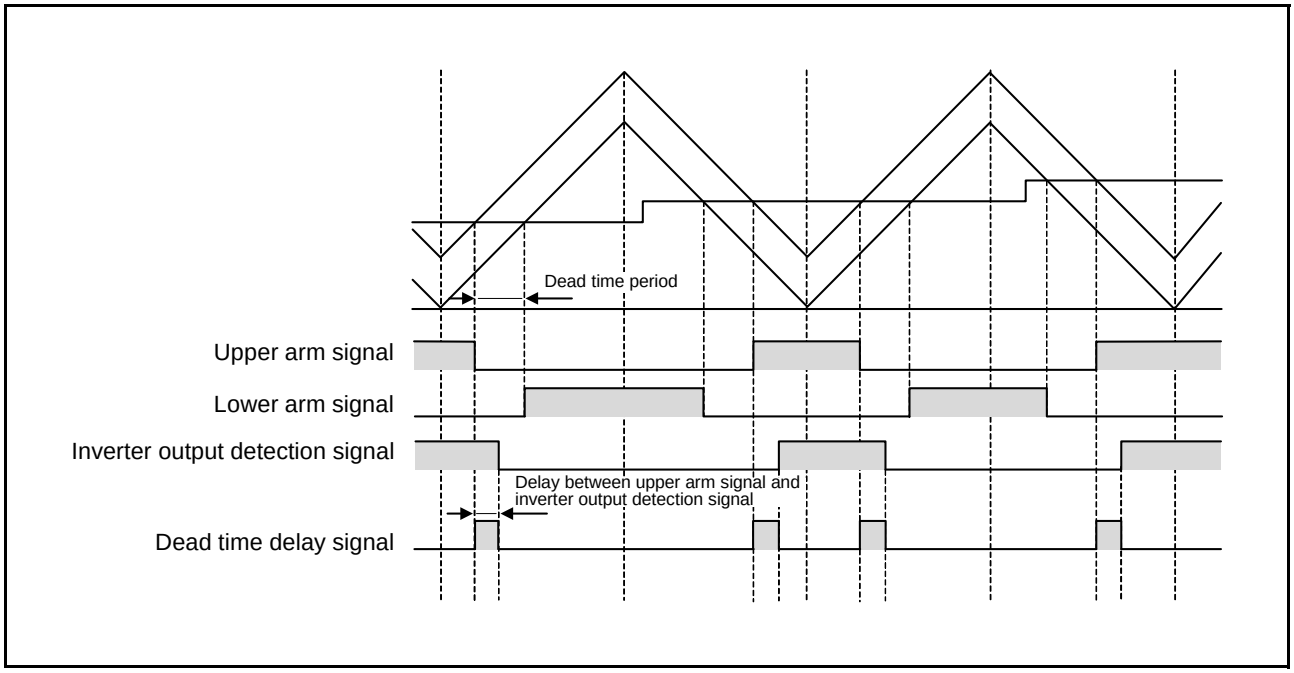
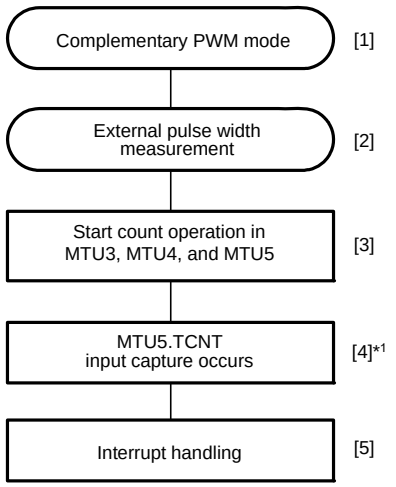



**Figure 20.79 Delay in Dead Time in Complementary PWM Mode Operation**

### (1) Example of Dead Time Compensation Setting Procedure

Figure 20.80 shows an example of dead time compensation setting procedure by using three counters in MTU5.




[1] Place MTU3 and MTU4 in complementary PWM mode. For details, refer to the Complementary PWM Mode section.

[2] Specify the external pulse width measurement function for the target TIOR in MTU5. For details, refer to the External Pulse Width Measurement section.

[3] Set bits CST3 and CST4 in MTU.TSTR and bits CSTU5, CSTV5, and CSTW5 in MTU5.TSTR to 1 to start count operation.

[4] When the capture condition specified in TIOR is satisfied, the MTU5.TCNT value is captured in MTU5.TGR.

[5] For U-phase dead time compensation, when an interrupt is generated at the crest (TGIA3) or trough (TCIV4) in complementary PWM mode, read the MTU5.TGRU value, calculate the difference in time between MTU5.TGRU and MTU3.TGRB, and write the corrected value to MTU3.TGRD in the interrupt handling.
For the V phase and W phase, read the MTU5.TGRV and MTU5.TGRW values and write the corrected values to MTU4.TGRC and MTU4.TGRD, respectively, in the same way as for U-phase compensation.
The MTU5.TCNT value should be cleared through the TCNTCMPCLR setting or by software.

Note 1. As an interrupt flag is set under the capture condition specified in TIOR, do not enable interrupt requests in MTU5.TIER.

**Figure 20.80 Example of Dead Time Compensation Setting Procedure**

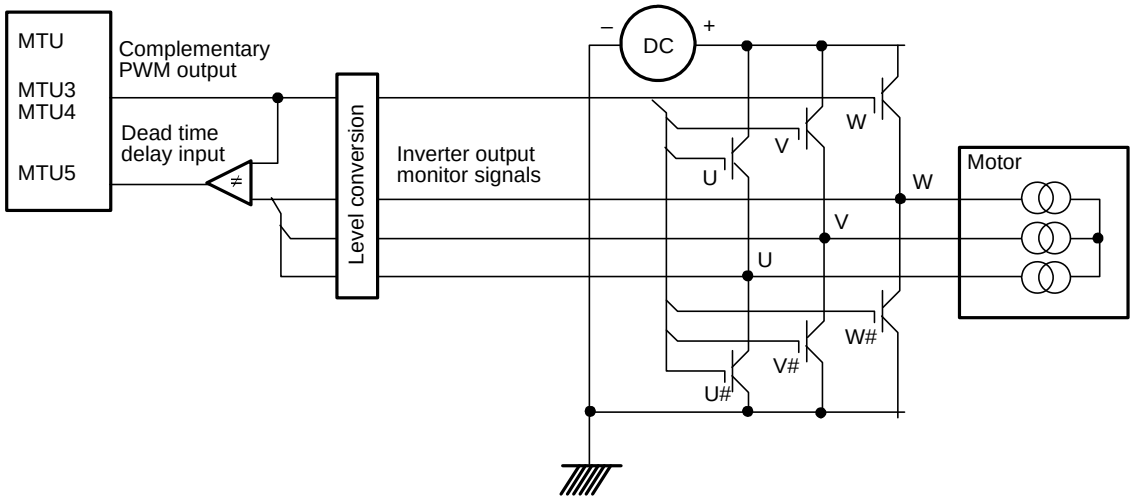



**Figure 20.81 Example of Motor Control Circuit Configuration**

### (2) TCNT Capture at Crest and/or Trough in Complementary PWM Mode Operation

The MTU5.TCNT value is captured in MTU5.TGR at either the crest or trough or at both the crest and trough during complementary PWM mode operation. The timing for capturing in MTU5.TGR can be selected by TIOR.
Figure 20.82 shows operation for the capture of MTU5.TCNT on crests and in troughs in complementary PWM mode.

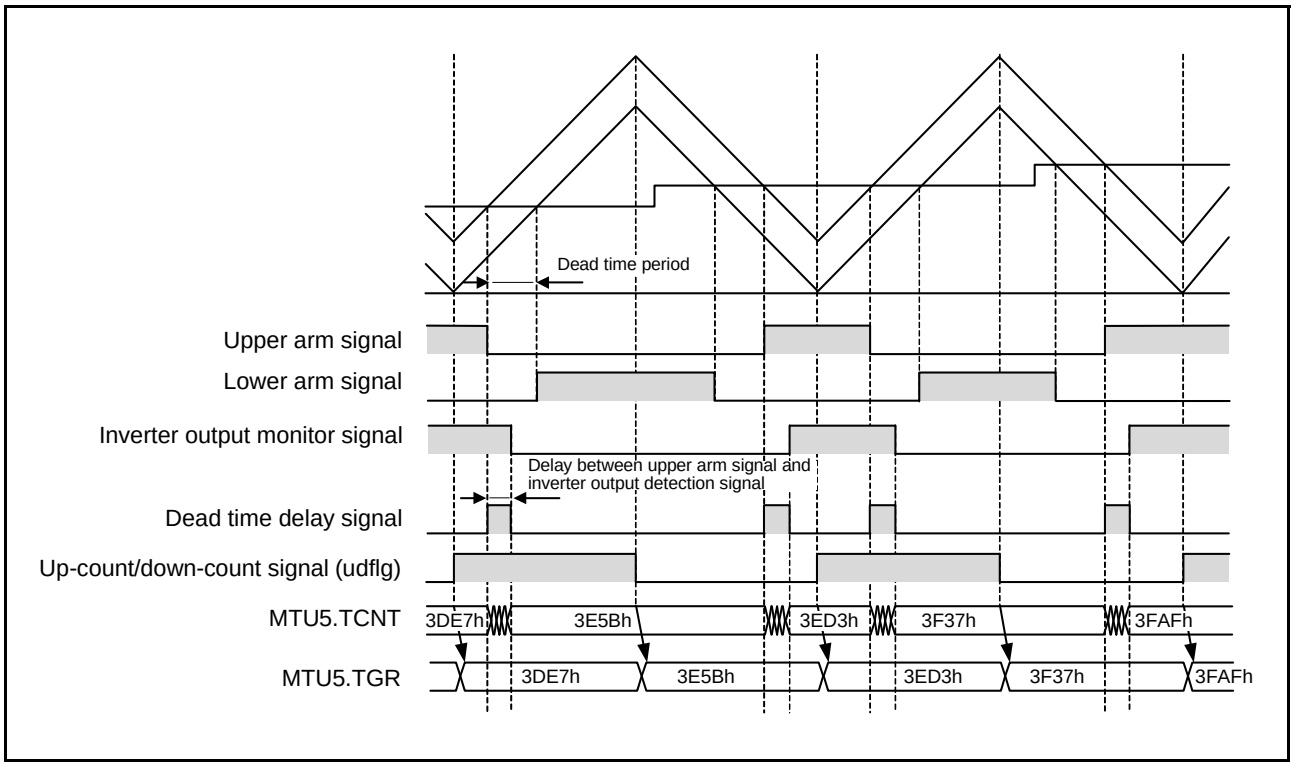



**Figure 20.82 MTU5.TCNT Capture at Crest and/or Trough in Complementary PWM Mode Operation**

### 20.3.12 Noise Filter

Each pin for use in input capture and external pulse input to the MTU is equipped with a noise filter. The noise filter samples input signals at the sampling clock and removes the pulses of which length is less than three sampling cycles. The noise filter functionality includes enabling and disabling of the noise filter for each pin and setting of the sampling clock for each channel. Figure 20.83 shows the timing of noise filtering.

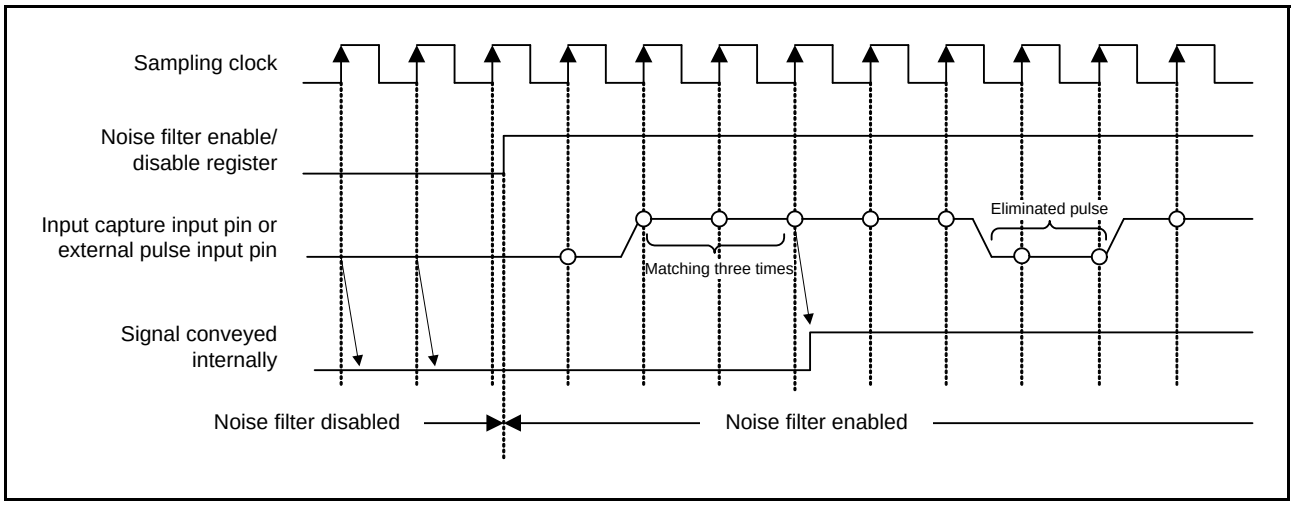



**Figure 20.83 Timing of Noise Filtering**

## 20.4 Interrupt Sources

### 20.4.1 Interrupt Sources and Priorities

There are three kinds of MTU interrupt source; TGR input capture/compare match, TCNT overflow, and TCNT underflow. Each interrupt source has its own enable/disable bit, allowing the generation of interrupt request signals to be enabled or disabled individually.

When an interrupt source is detected, an interrupt is requested if the corresponding enable/disable bit in TIER is set to 1. Relative channel priorities can be changed by the interrupt controller; however the priority within a channel is fixed. For details, refer to section 14, Interrupt Controller (ICUb).

Table 20.57 lists the MTU interrupt sources.

**Table 20.57 MTU Interrupt Sources (1)**

| Channel | Name | Interrupt Source | DTC Activation | Priority |
|---|---|---|---|---|
| MTU0 | TGIA0 | MTU0.TGRA input capture/compare match | Possible | High |
| | TGIB0 | MTU0.TGRB input capture/compare match | Possible | |
| | TGIC0 | MTU0.TGRC input capture/compare match | Possible | |
| | TGID0 | MTU0.TGRD input capture/compare match | Possible | |
| | TCIV0 | MTU0.TCNT overflow | Not possible | |
| | TGIE0 | MTU0.TGRE compare match | Not possible | |
| | TGIF0 | MTU0.TGRF compare match | Not possible | |
| MTU1 | TGIA1 | MTU1.TGRA input capture/compare match | Possible | |
| | TGIB1 | MTU1.TGRB input capture/compare match | Possible | |
| | TCIV1 | MTU1.TCNT overflow | Not possible | |
| | TCIU1 | MTU1.TCNT underflow | Not possible | |
| MTU2 | TGIA2 | MTU2.TGRA input capture/compare match | Possible | |
| | TGIB2 | MTU2.TGRB input capture/compare match | Possible | |
| | TCIV2 | MTU2.TCNT overflow | Not possible | |
| | TCIU2 | MTU2.TCNT underflow | Not possible | |
| MTU3 | TGIA3 | MTU3.TGRA input capture/compare match | Possible | |
| | TGIB3 | MTU3.TGRB input capture/compare match | Possible | |
| | TGIC3 | MTU3.TGRC input capture/compare match | Possible | |
| | TGID3 | MTU3.TGRD input capture/compare match | Possible | |
| | TCIV3 | MTU3.TCNT overflow | Not possible | |
| MTU4 | TGIA4 | MTU4.TGRA input capture/compare match | Possible | |
| | TGIB4 | MTU4.TGRB input capture/compare match | Possible | |
| | TGIC4 | MTU4.TGRC input capture/compare match | Possible | |
| | TGID4 | MTU4.TGRD input capture/compare match | Possible | |
| | TCIV4 | MTU4.TCNT overflow/underflow | Possible | |
| MTU5 | TGIU5 | MTU5.TGRU input capture/compare match | Possible | |
| | TGIV5 | MTU5.TGRV input capture/compare match | Possible | |
| | TGIW5 | MTU5.TGRW input capture/compare match | Possible | Low |

Note: • This table lists the initial state immediately after a reset. The relative channel priorities can be changed by the interrupt controller.

#### (1) Input Capture/Compare Match Interrupt

An interrupt is requested if the TIER.TGIE bit is set to 1 when a TGR input capture/compare match occurs on a channel. The MTU has 21 input capture/compare match interrupts (six for MTU0, four each for MTU3 and MTU4, two each for MTU1 and MTU2, and three for MTU5).

#### (2) Overflow Interrupt

An interrupt is requested if the TIER.TGIE bit is set to 1 when a TCNT overflow occurs on a channel. The MTU has five overflow interrupts (one for each channel).

#### (3) Underflow Interrupt

An interrupt is requested if the TIER.TCIEU bit is set to 1 when a TCNT underflow occurs on a channel. The MTU has two underflow interrupts (one each for MTU1 and MTU2).

### 20.4.2 DTC Activation

The DTC can be activated by the TGR input capture/compare match interrupt in each channel or the overflow interrupt in MTU4. For details, refer to section 16, Data Transfer Controller (DTCa).
The MTU provides a total of 20 input capture/compare match interrupts and overflow interrupts that can be used as DTC activation sources: four each for MTU0 and MTU3, two each for MTU1 and MTU2, five for MTU4, and three for MTU5.

### 20.4.3 A/D Converter Activation

The A/D converter can be activated by one of the following five methods in the MTU. Table 20.58 lists the relationship between interrupt sources and A/D converter start request signals.

#### (1) A/D Converter Activation by TGRA Input Capture/Compare Match or at MTU4.TCNT Trough in Complementary PWM Mode

The A/D converter can be activated by the occurrence of a TGRA input capture/compare match in each channel. In addition, if complementary PWM mode operation is performed while the MTU4.TIER.TTGE2 bit is set to 1, the A/D converter can be activated at the trough of MTU4.TCNT count (MTU4.TCNT = 0000h).
A/D converter start request signal TRGAN is issued to the A/D converter under either of the following conditions.

- When a TGRA input capture/compare match occurs on a channel while the TIER.TTGE bit is set to 1
- When the MTU4.TCNT count reaches the trough (MTU4.TCNT = 0000h) during complementary PWM mode operation while the MTU4.TIER.TTGE2 bit is set to 1

When either condition is satisfied, if A/D converter start signal TRGAN from the MTU is selected as the trigger in the A/D converter, A/D conversion will start.

#### (2) A/D Converter Activation by Compare Match between MTU0.TCNT and MTU0.TGRE

A compare match between MTU0.TCNT and MTU0.TGRE activates the A/D converter.
A/D converter start request signal TRG0EN is issued when a compare match occurs between MTU0.TCNT and MTU0.TGRE. If A/D converter start signal TRG0EN from the MTU is selected as the trigger in the A/D converter, A/D conversion will start.

#### (3) A/D Converter Activation by Compare Match between MTU0.TCNT and MTU0.TGRF

An input capture or compare match between MTU0.TCNT, MTU0.TGRA, and MTU0.TGRB activates the A/D converter.
A compare match between MTU0.TCNT and MTU0.TGRF activates the A/D converter.
A/D converter start request signal TRG0FN is issued when a compare match occurs between MTU0.TCNT and MTU0.TGRF. If A/D converter start signal TRG0FN from the MTU is selected as the trigger in the A/D converter, A/D conversion will start.

### (4) A/D Converter Activation by Input Capture or Compare Match with MTU0.TGRA or TGRB

The A/D converter can be activated when an input capture or compare match occurs between MTU0.TCNT and MTU0.TGRA or MTU0.TGRB.

When an input capture or compare match occurs between MTU0.TCNT and MTU0.TGRA or MTU0.TGRB. A/D converter start request signal TRG0AN or TRG0BN is issued. If A/D converter start signal TRG0AN or TRG0BN from the MTU is selected as the trigger in the A/D converter, A/D conversion will start.

### (5) A/D Converter Activation by A/D Converter Start Request Delaying Function

The A/D converter can be activated by generating A/D converter start request signal TRG4AN or TRG4BN when the MTU4.TCNT count matches the TADCORA or TADCORB value if the TADCR.UT4AE bit, TADCR.DT4AE bit, TADCR.UT4BE bit, or TADCR.DT4BE bit is set to 1. For details, refer to section 20.3.9, A/D Converter Start Request Delaying Function.

A/D conversion will start if A/D converter start signal TRG4ABN from the MTU is selected as the trigger in the A/D converter when TRG4AN or TRG4BN is generated.

**Table 20.58 Interrupt Sources and A/D Converter Start Request Signals**

| Target Registers | A/D Start Request Source | A/D Converter Start Request Signal |
|---|---|---|
| MTU0.TGRA and MTU0.TCNT | Input capture/compare match | TRGAN |
| MTU1.TGRA and MTU1.TCNT | | |
| MTU2.TGRA and MTU2.TCNT | | |
| MTU3.TGRA and MTU3.TCNT | | |
| MTU4.TGRA and MTU4.TCNT | | |
| MTU4.TCNT | MTU4.TCNT trough in complementary PWM mode | |
| MTU0.TGRA and MTU0.TCNT | Input capture/compare match | TRG0AN |
| MTU0.TGRB and MTU0.TCNT | | TRG0BN |
| MTU0.TGRE and MTU0.TCNT | Compare match | TRG0EN |
| MTU0.TGRF and MTU0.TCNT | | TRG0FN |
| TADCORA and MTU4.TCNT | | TRG4AN |
| TADCORB and MTU4.TCNT | | TRG4BN |
| TADCORA and MTU4.TCNT or TADCORB and MTU4.TCNT | | TRG4ABN |

## 20.5 Operation Timing

### 20.5.1 Input/Output Timing

#### (1) TCNT Count Timing

Figure 20.84 and Figure 20.85 show the TCNT count timing for TGI interrupt in internal clock operation, Figure 20.86 shows the TCNT count timing in external clock operation (normal mode), and Figure 20.87 shows the TCNT count timing in external clock operation (phase counting mode).

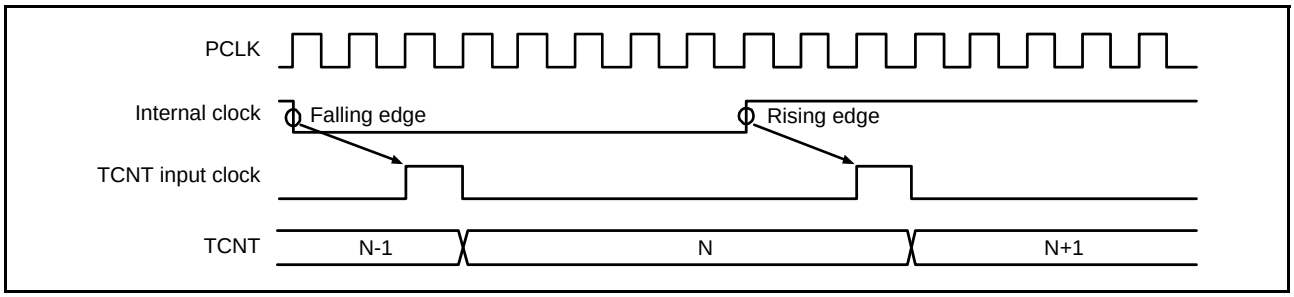



**Figure 20.84 Count Timing in Internal Clock Operation (MTU0 to MTU4)**

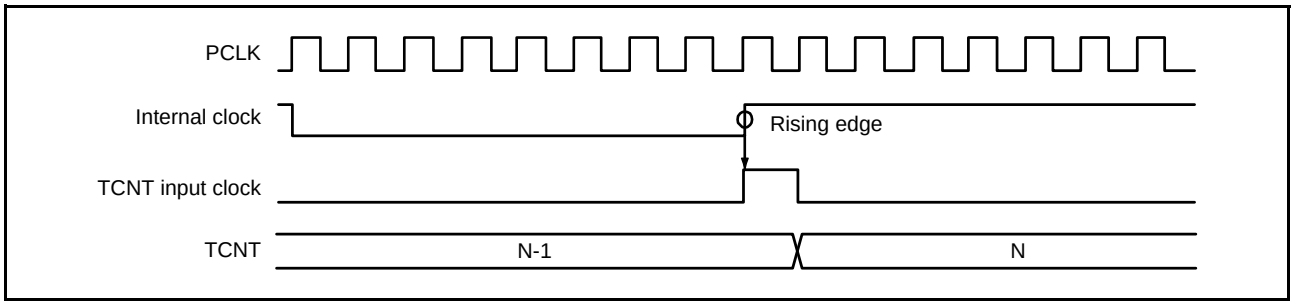



**Figure 20.85 Count Timing in Internal Clock Operation (MTU5)**

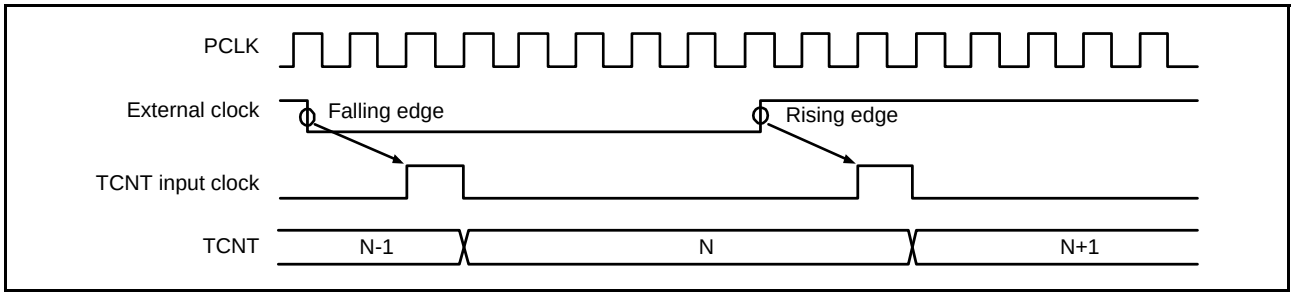



**Figure 20.86 Count Timing in External Clock Operation (MTU0 to MTU4)**

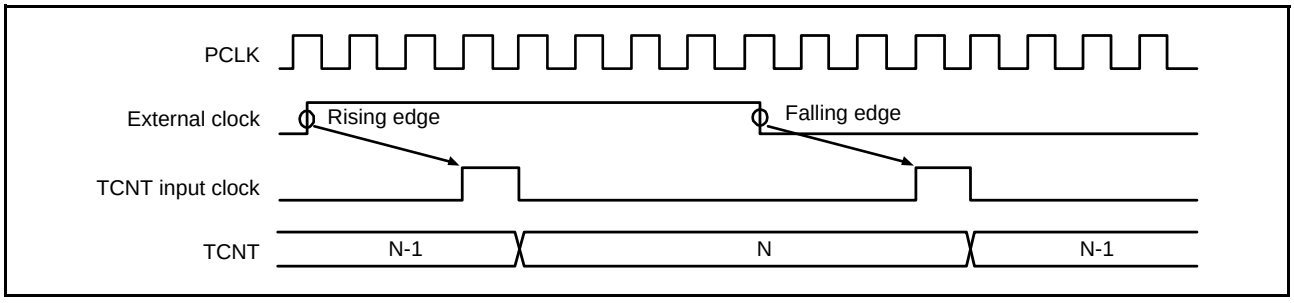



**Figure 20.87 Count Timing in External Clock Operation (Phase Counting Mode)**

### (2) Output Compare Output Timing

A compare match signal is generated in the final state in which TCNT and TGR match (the point at which the count value matched is updated by TCNT). When a compare match signal is generated, the value set in TIOR is output to the output compare output pin (MTIOC pin). After a match between TCNT and TGR, the compare match signal is not generated until the TCNT input clock is generated.

Figure 20.88 shows the output compare output timing (normal mode or PWM mode) and Figure 20.89 shows the output compare output timing (complementary PWM mode or reset-synchronized PWM mode).

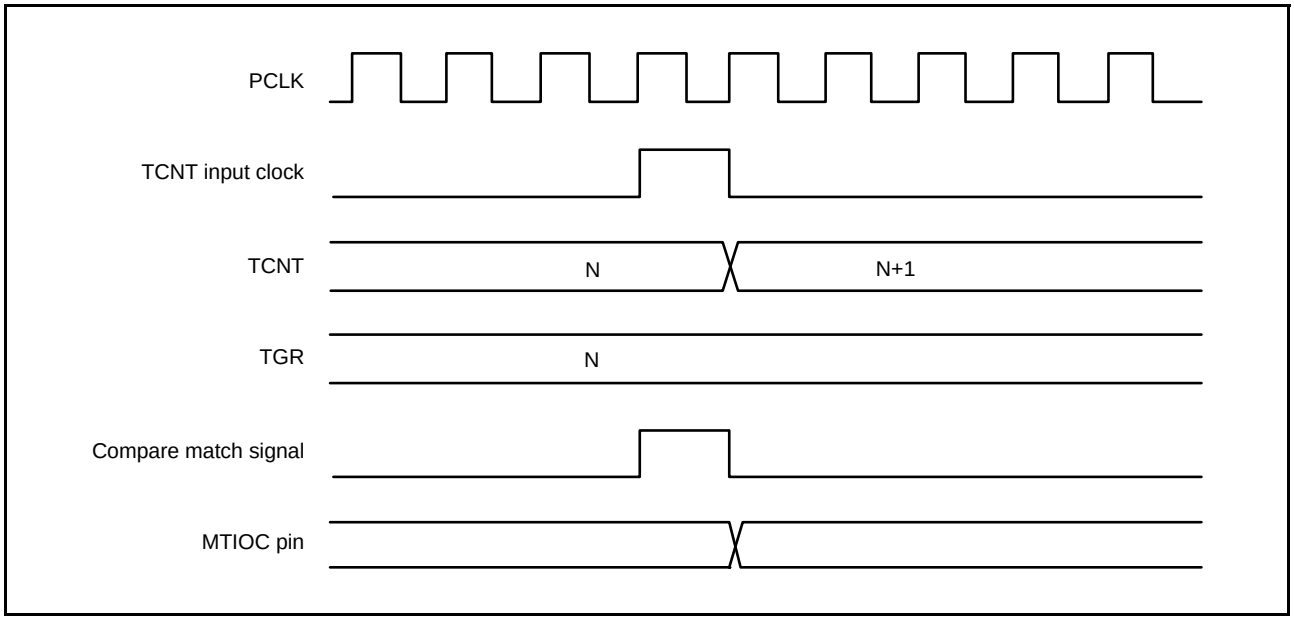



**Figure 20.88 Output Compare Output Timing (Normal Mode or PWM Mode)**

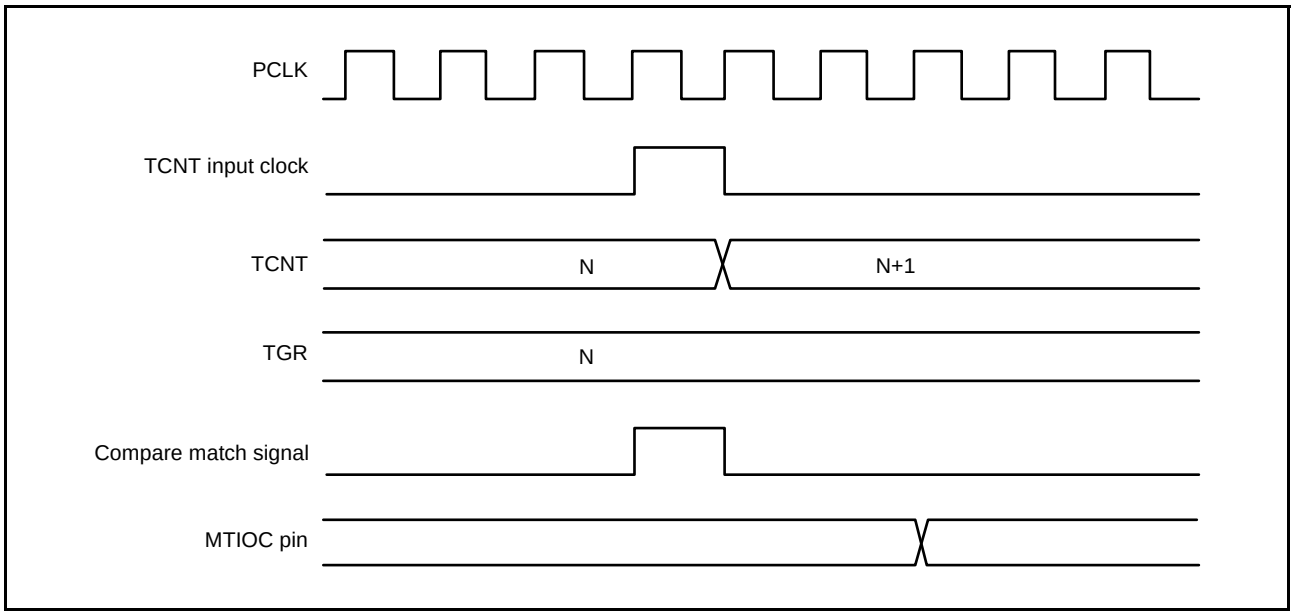



**Figure 20.89 Output Compare Output Timing (Complementary PWM Mode or Reset-Synchronized PWM Mode)**

### (3) Input Capture Signal Timing

Figure 20.90 shows the input capture signal timing.

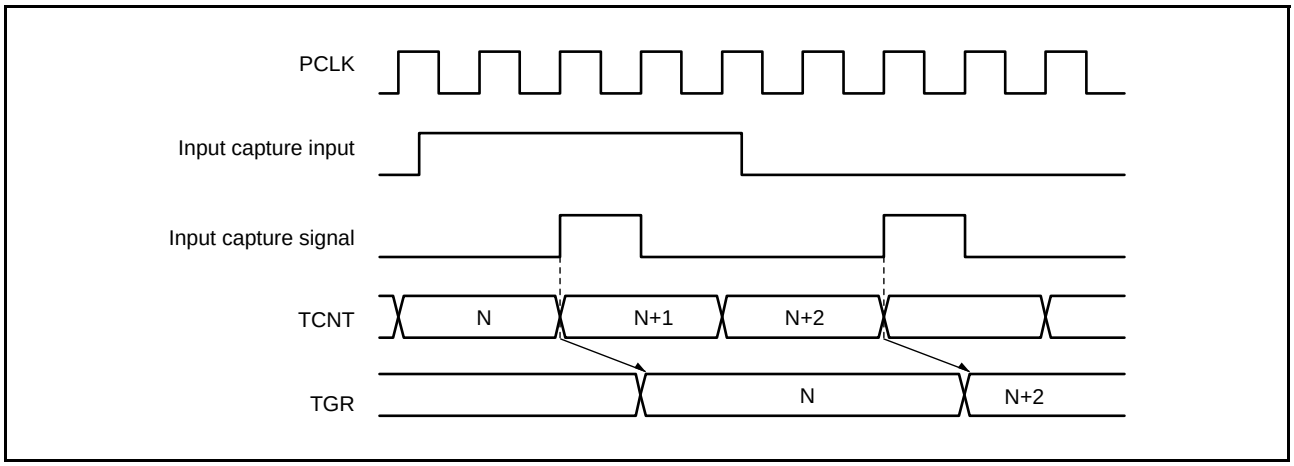



**Figure 20.90 Input Capture Input Signal Timing**

### (4) Timing for Counter Clearing by Compare Match/Input Capture

Figure 20.91 and Figure 20.92 show the timing when counter clearing on compare match is specified, and Figure 20.93 shows the timing when counter clearing on input capture is specified.

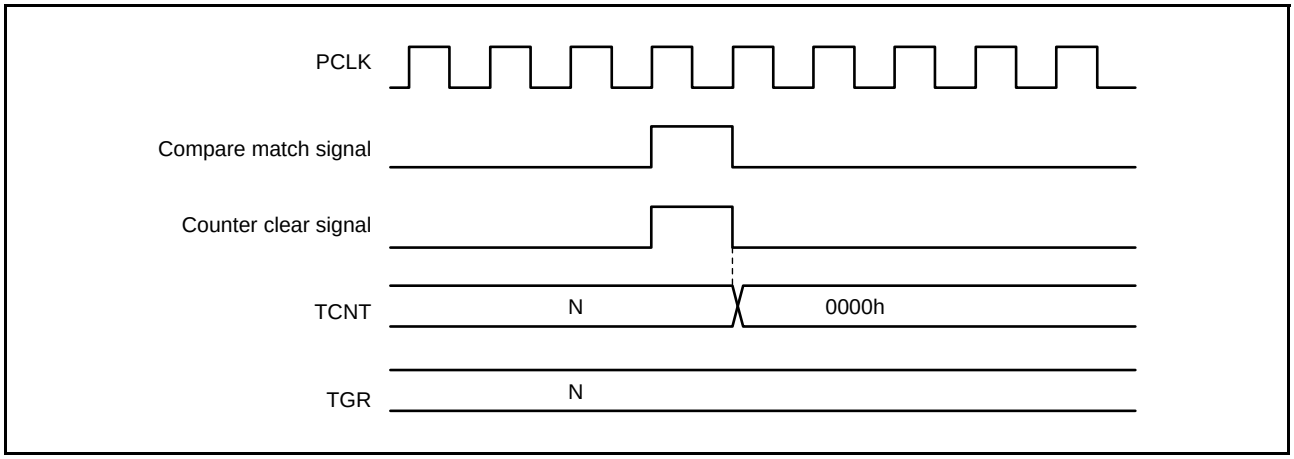



**Figure 20.91 Counter Clear Timing (Compare Match) (MTU0 to MTU4)**

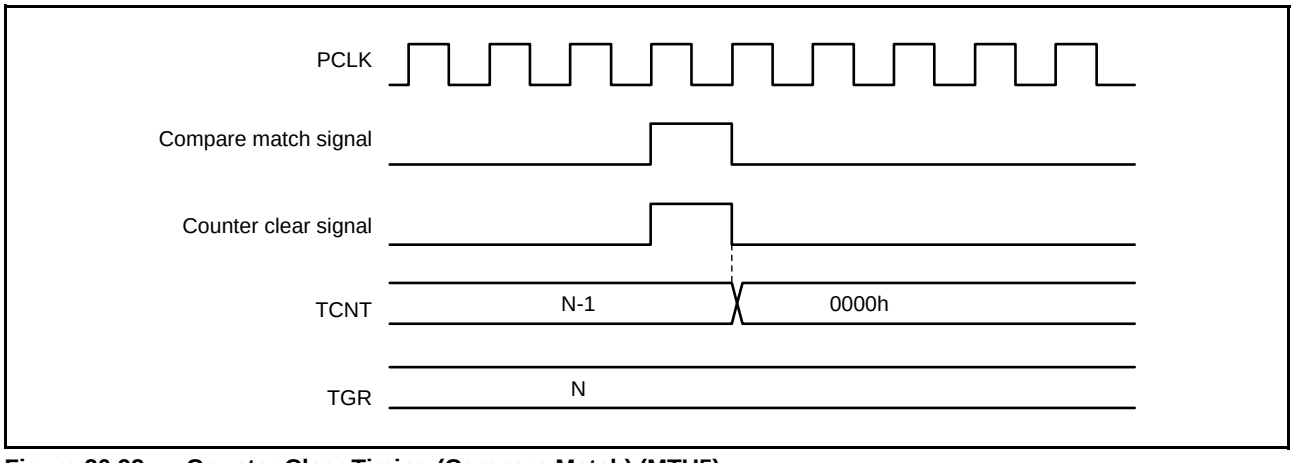



**Figure 20.92 Counter Clear Timing (Compare Match) (MTU5)**

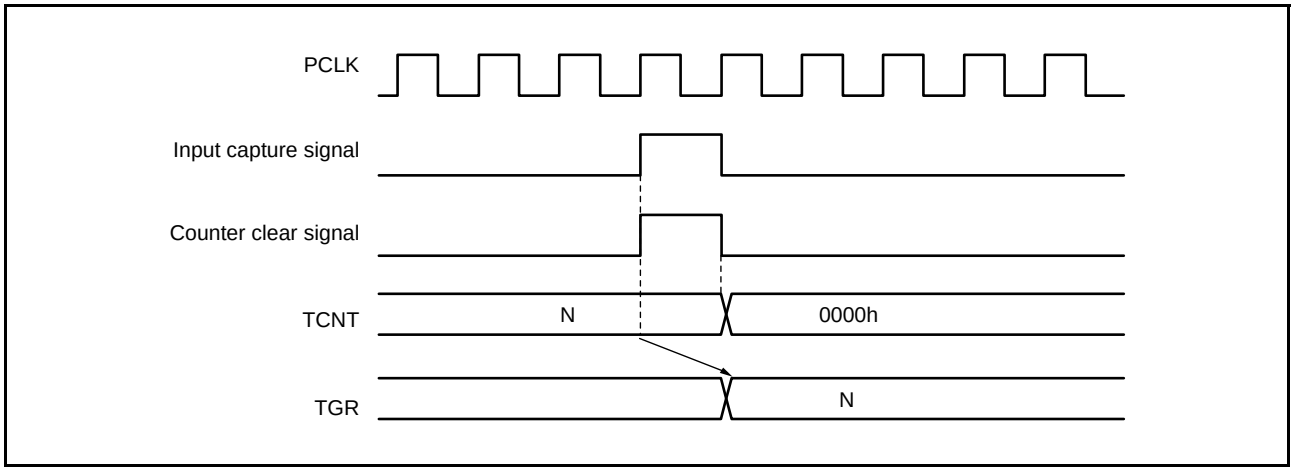



**Figure 20.93 Counter Clear Timing (Input Capture) (MTU0 to MTU5)**

### (5) Buffer Operation Timing

Figure 20.94 to Figure 20.96 show the timing in buffer operation.

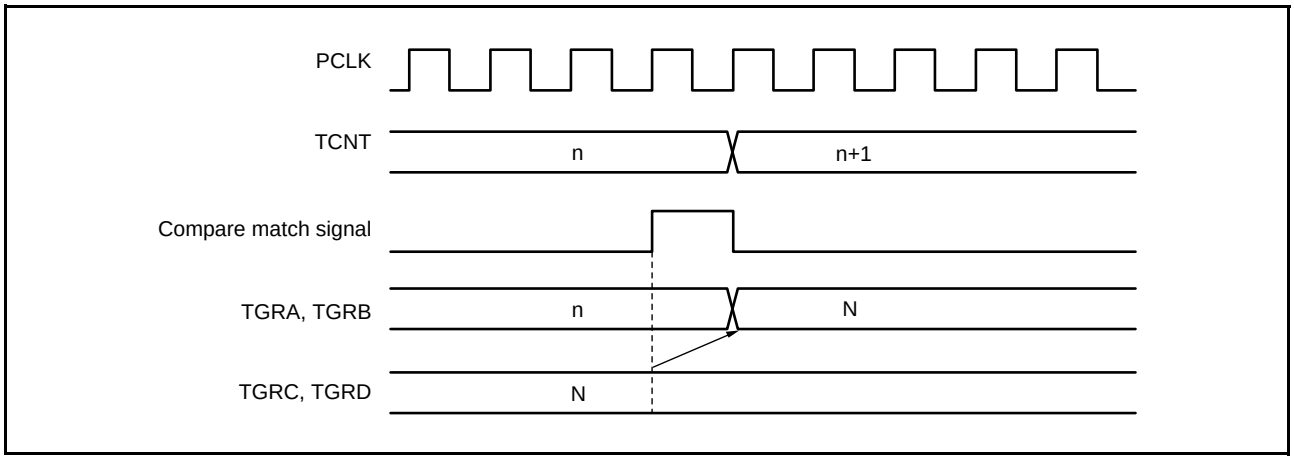



**Figure 20.94 Buffer Operation Timing (Compare Match)**

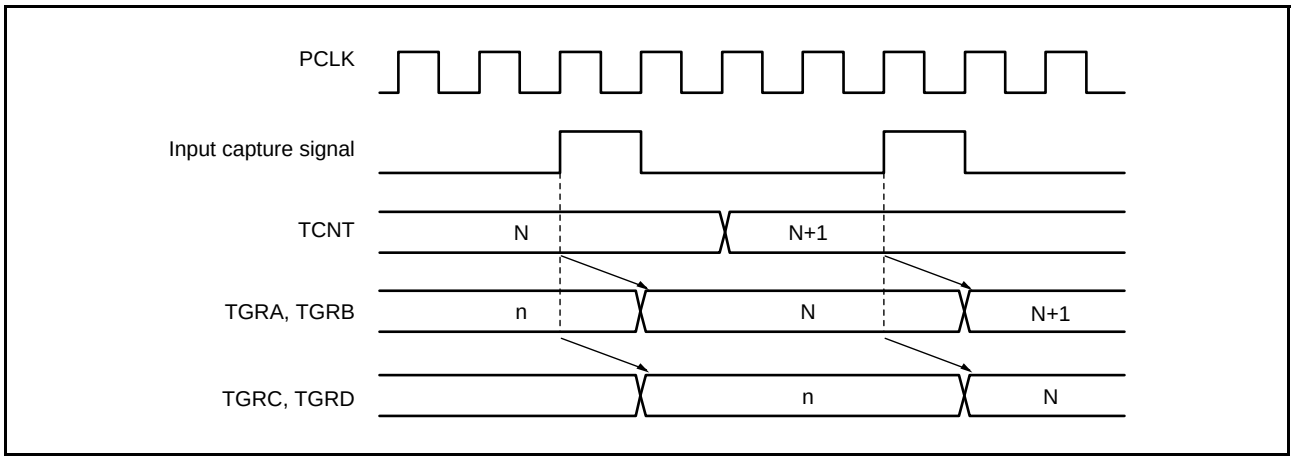



**Figure 20.95 Buffer Operation Timing (Input Capture)**

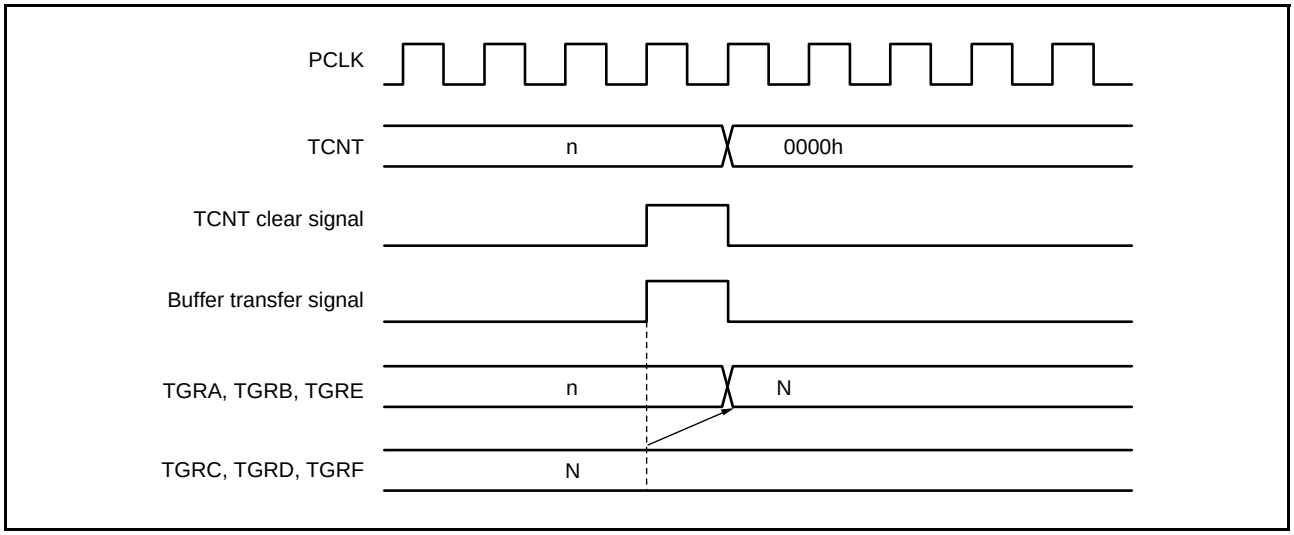



**Figure 20.96 Buffer Operation Timing (When TCNT Cleared)**

### (6) Buffer Transfer Timing (Complementary PWM Mode)

Figure 20.97 to Figure 20.99 show the buffer transfer timing in complementary PWM mode.

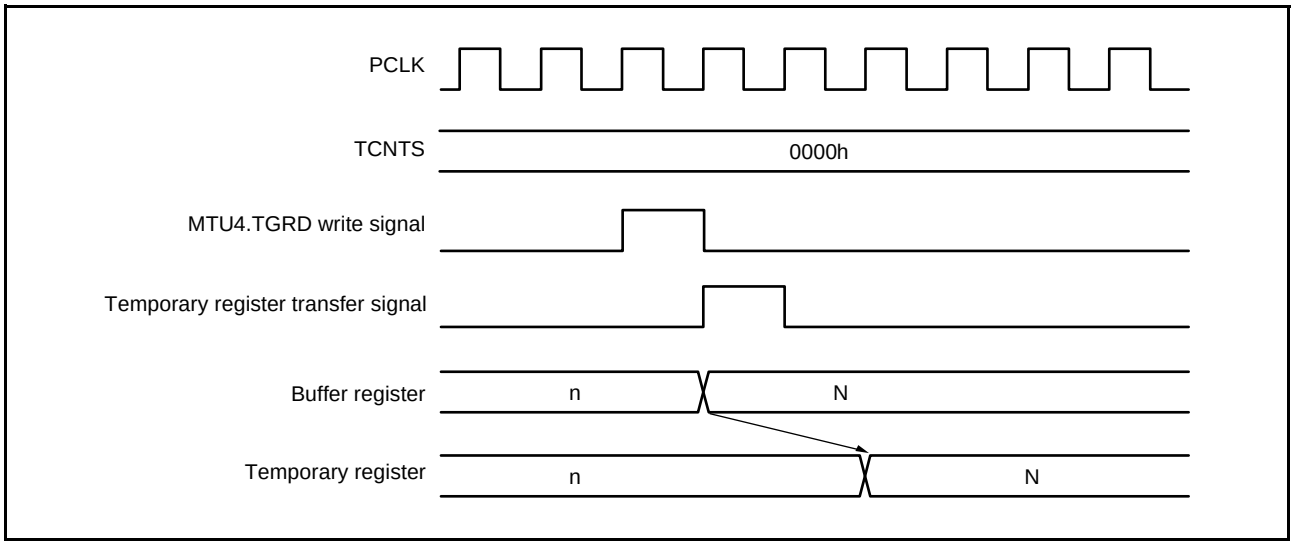



**Figure 20.97 Transfer Timing from Buffer Register to Temporary Register (TCNTS Stop)**

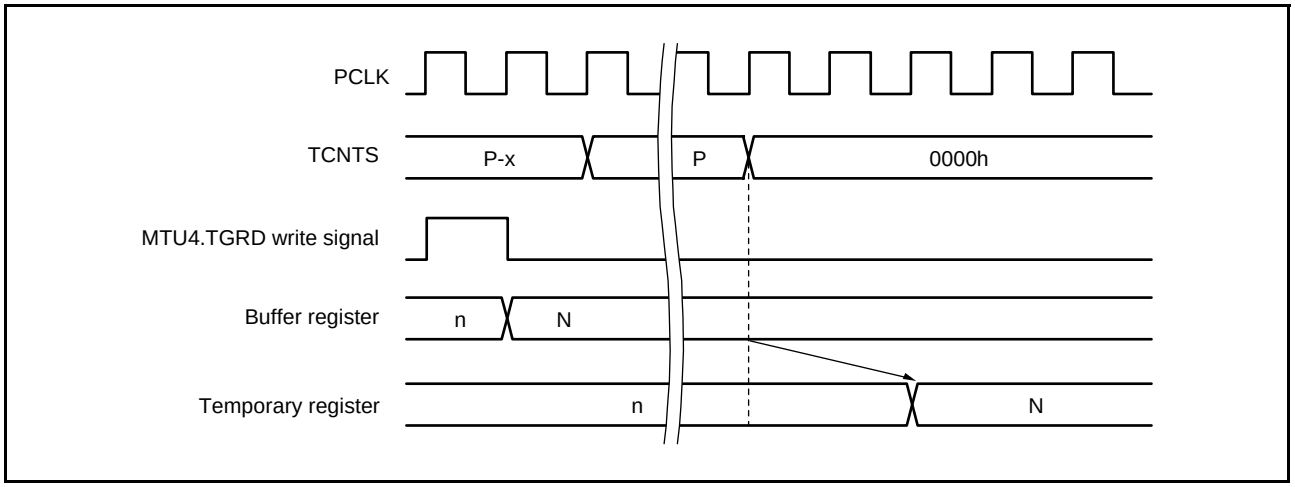



**Figure 20.98 Transfer Timing from Buffer Register to Temporary Register (TCNTS Operating)**

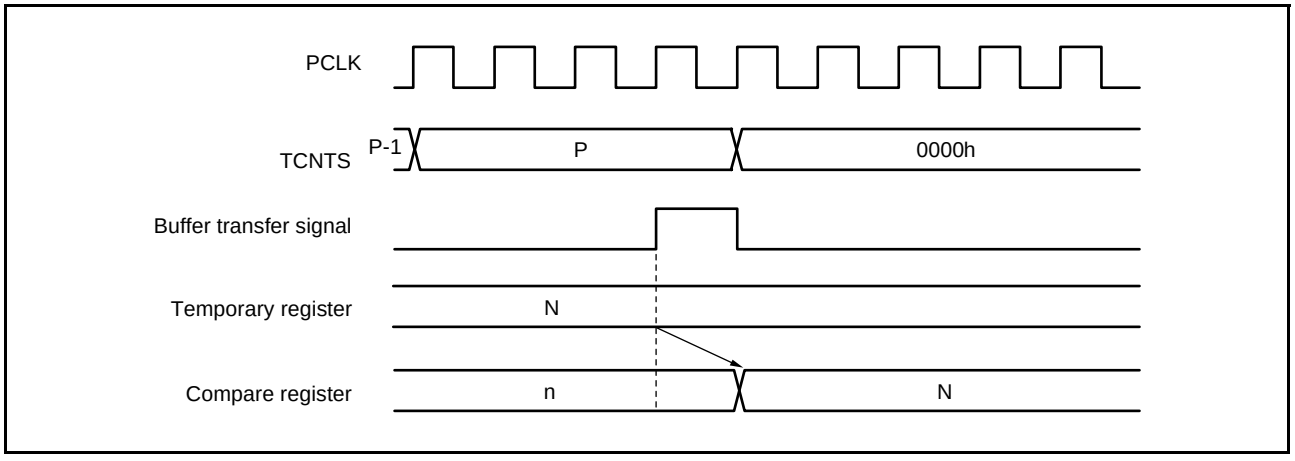



**Figure 20.99 Transfer Timing from Temporary Register to Compare Register**

## 20.5.2 Interrupt Signal Timing

### (1) Timing for TGI Interrupt by Compare Match

Figure 20.100 and Figure 20.101 show the TGI interrupt request signal timing on compare match.

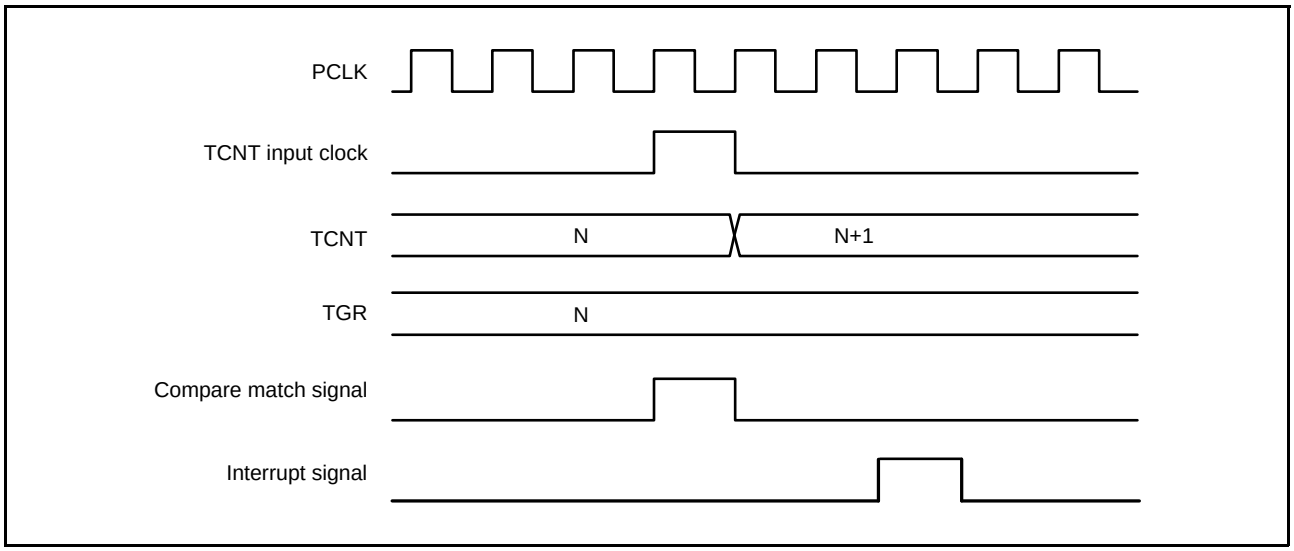



**Figure 20.100 TGI Interrupt Timing (Compare Match) (MTU0 to MTU4)**

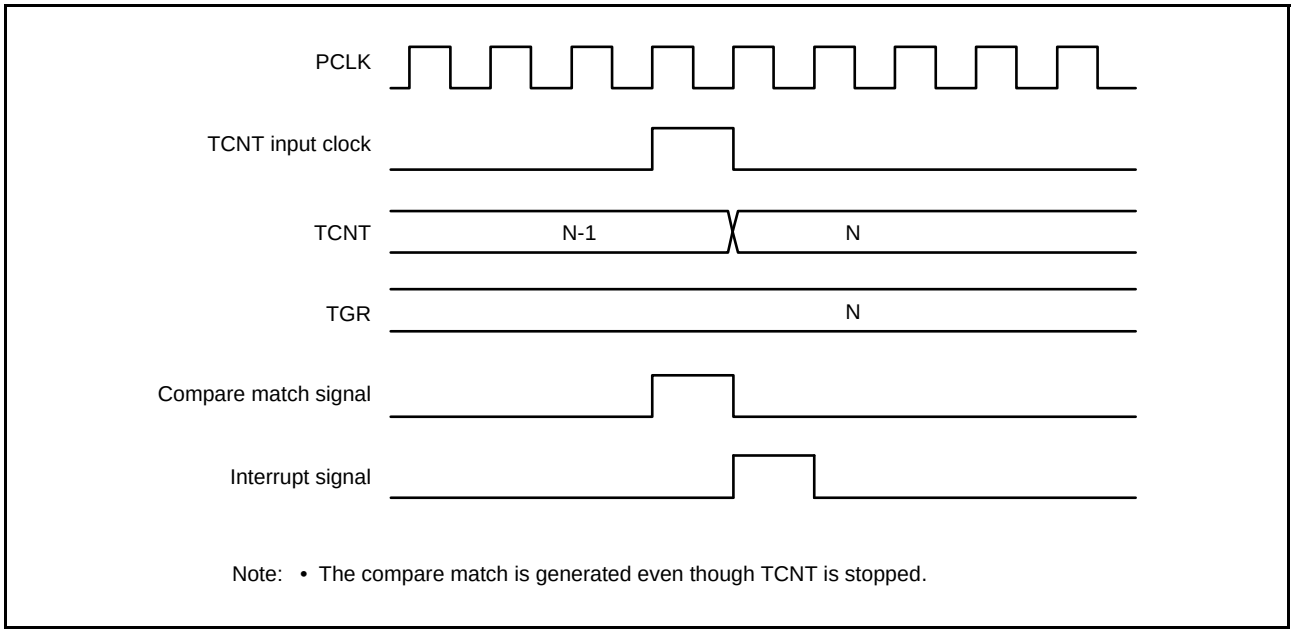



**Figure 20.101 TGI Interrupt Timing (Compare Match) (MTU5)**

### (2) Timing for TGI Interrupt by Input Capture

Figure 20.102 and Figure 20.103 show TGI interrupt request signal timing on input capture.

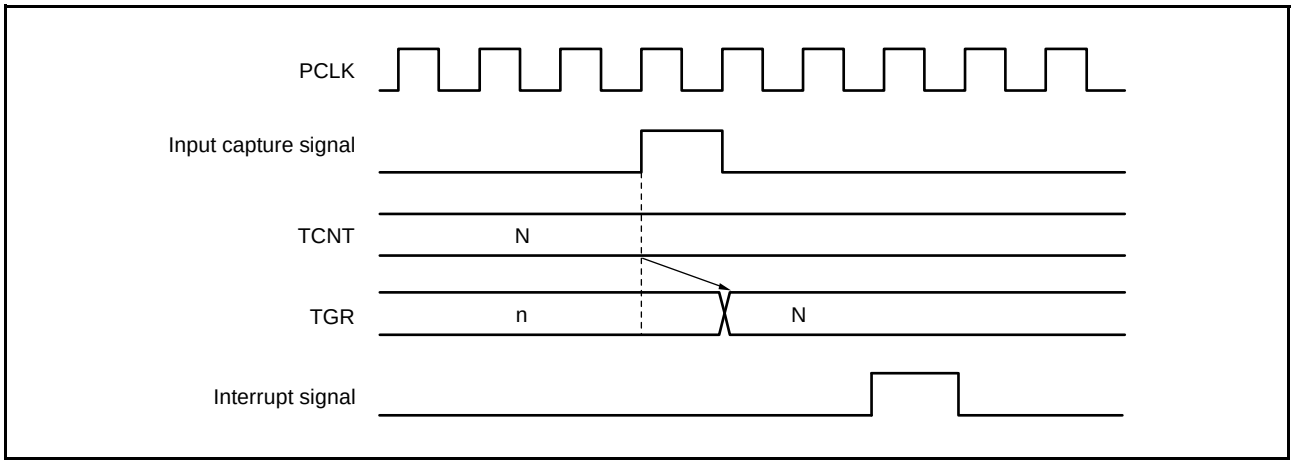



**Figure 20.102 TGI Interrupt Timing (Input Capture) (MTU0 to MTU4)**

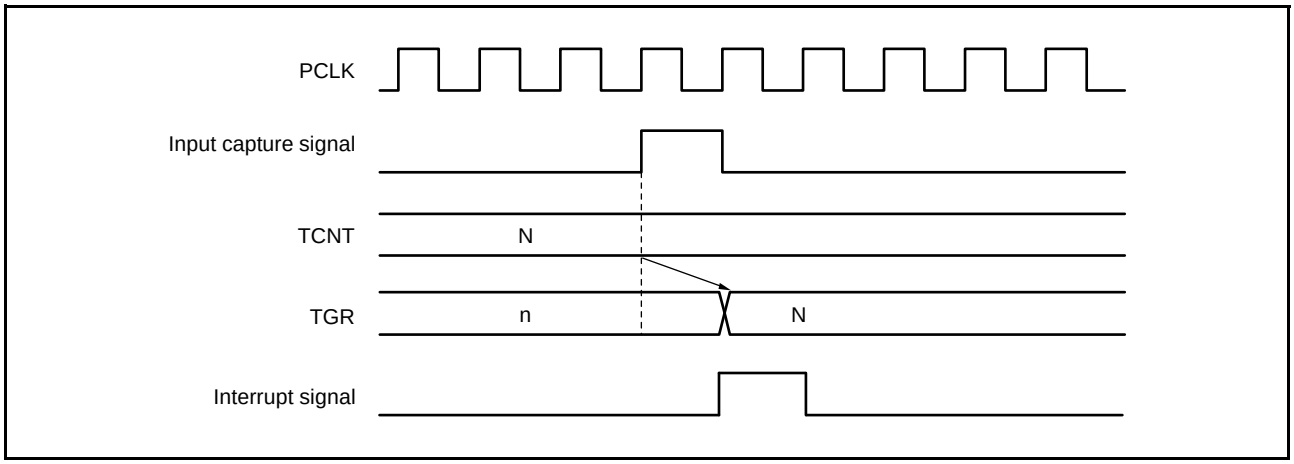



**Figure 20.103 TGI Interrupt Timing (Input Capture) (MTU5)**

### (3) TCIV and TCIU Interrupt Timing

Figure 20.104 shows the TCIV interrupt request signal timing on overflow.
Figure 20.105 shows the TCIU interrupt request signal timing on underflow.

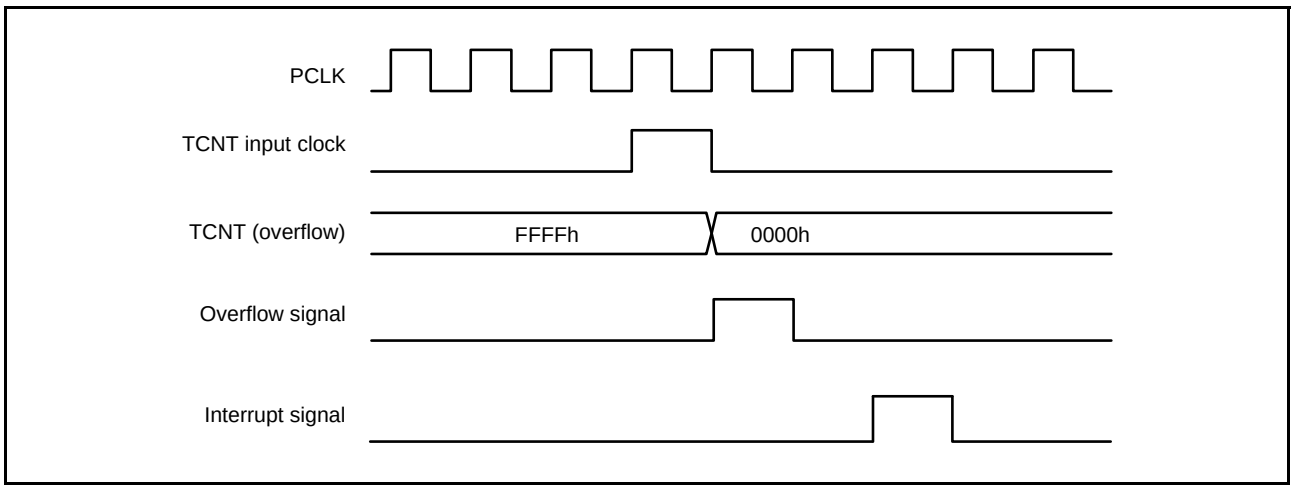



**Figure 20.104 TCIV Interrupt Timing**

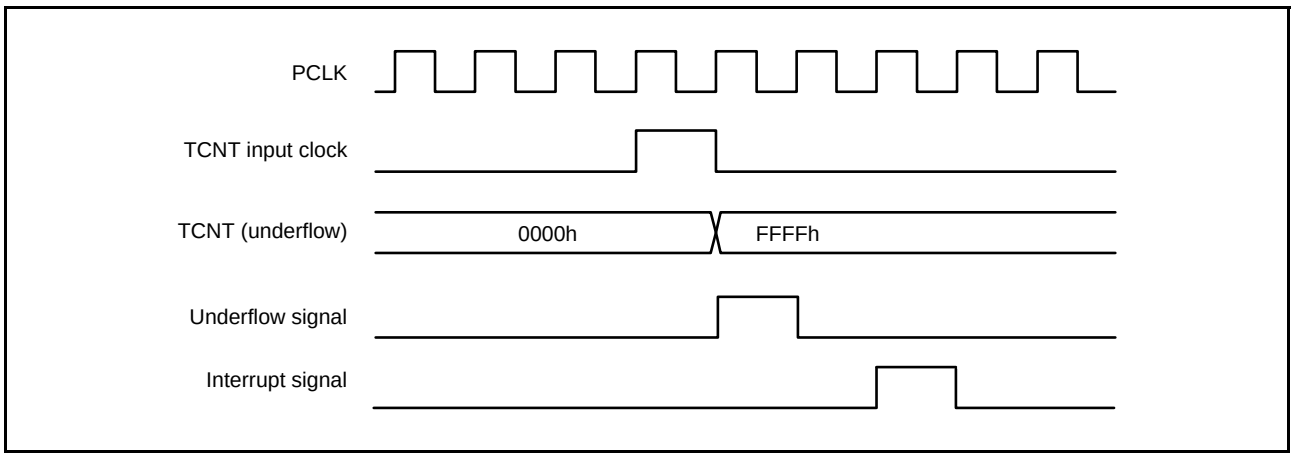



**Figure 20.105 TCIU Interrupt Timing**

## 20.6 Usage Notes

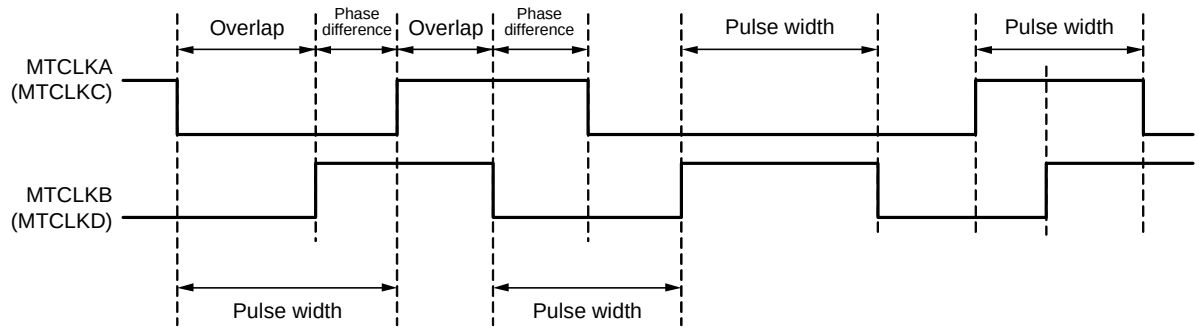
### 20.6.1 Module Clock Stop Mode Setting

MTU operation can be disabled or enabled using the module stop control register. MTU operation is stopped with the initial setting. Register access is enabled by clearing the module clock stop mode. For details, refer to section 11, Low Power Consumption.

### 20.6.2 Input Clock Restrictions

The input clock pulse width must be at least 1.5 PCLK clock for single-edge detection, and at least 2.5 PCLK clock for both-edge detection. The MTU will not operate properly at narrower pulse widths.

In phase counting mode, the phase difference and overlap between the two input clocks must be at least 1.5 PCLK clock, and the pulse width must be at least 2.5 PCLK clock. Figure 20.106 shows the input clock conditions in phase counting mode.




Note: • Phase difference and overlap: 1.5 PCLK clock or more
Pulse width: 2.5 PCLK clock or more

**Figure 20.106 Phase Difference, Overlap, and Pulse Width in Phase Counting Mode**

### 20.6.3 Note on Cycle Setting

When counter clearing on compare match is set, TCNT is cleared in the final state in which it matches the TGR value (the point at which TCNT updates the matched count value). Consequently, the actual counter frequency is given by the following formula:

- MTU0 to MTU4

$$f = \frac{CNTCLK}{(N + 1)}$$

- MTU5

$$f = \frac{CNTCLK}{N}$$

f: Counter frequency
CNTCLK: The counter-clock frequency set by TCR.TPSC[2:0] bits
N: TGR setting

### 20.6.4 Contention between TCNT Write and Clear Operations

If the counter clear signal is generated in a TCNT write cycle, TCNT clearing takes precedence and TCNT write operation is not performed.
Figure 20.107 shows the timing in this case.

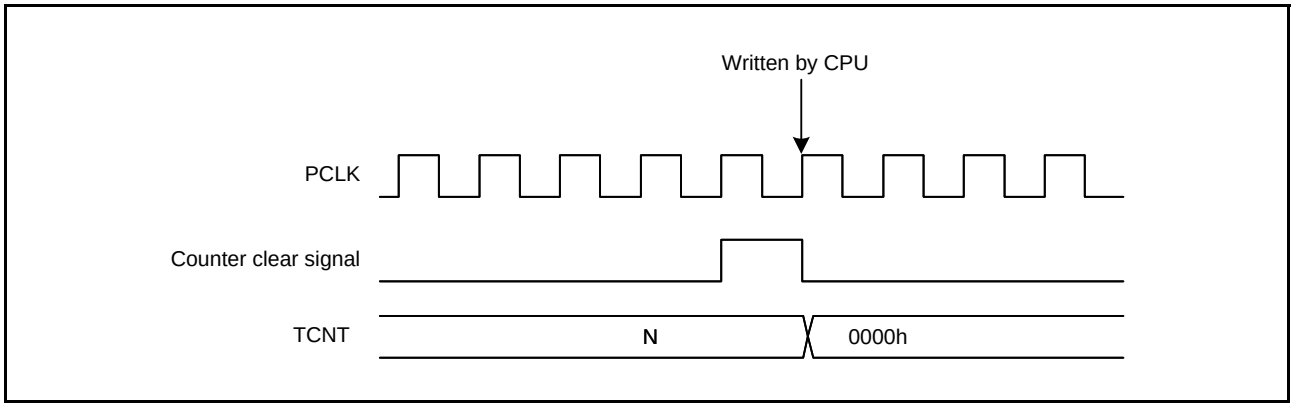



**Figure 20.107 Contention between TCNT Write and Counter Clear Operations**

### 20.6.5 Contention between TCNT Write and Increment Operations

If incrementing occurs in a TCNT write cycle, TCNT write operation takes precedence and TCNT is not incremented. Figure 20.108 shows the timing in this case.

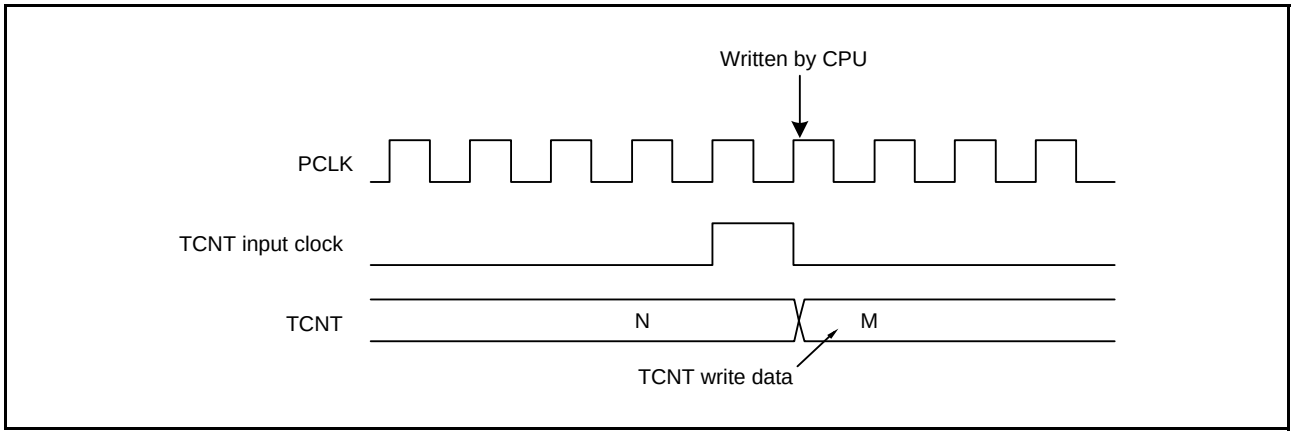



**Figure 20.108 Contention between TCNT Write and Increment Operations**

### 20.6.6 Contention between TGR Write Operation and Compare Match

If a compare match occurs in a TGR write cycle, TGR write operation is executed and the compare match signal is also generated.
Figure 20.109 shows the timing in this case.

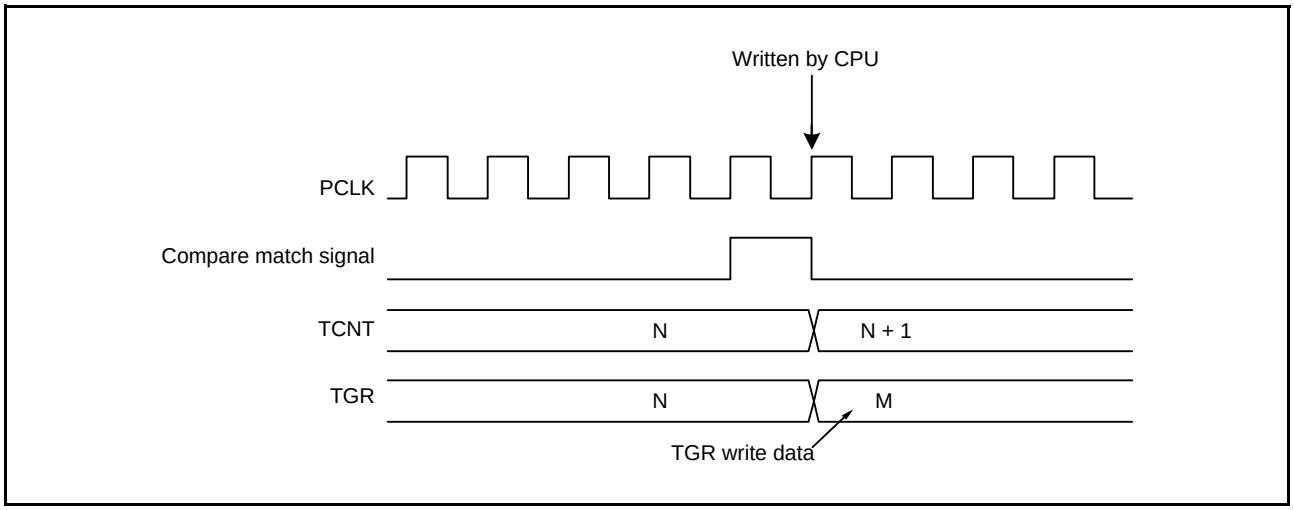



**Figure 20.109 Contention between TGR Write Operation and Compare Match**

### 20.6.7 Contention between Buffer Register Write Operation and Compare Match

If a compare match occurs in a TGR write cycle, the data before write operation is transferred to TGR by the buffer operation.

Figure 20.110 shows the timing in this case.

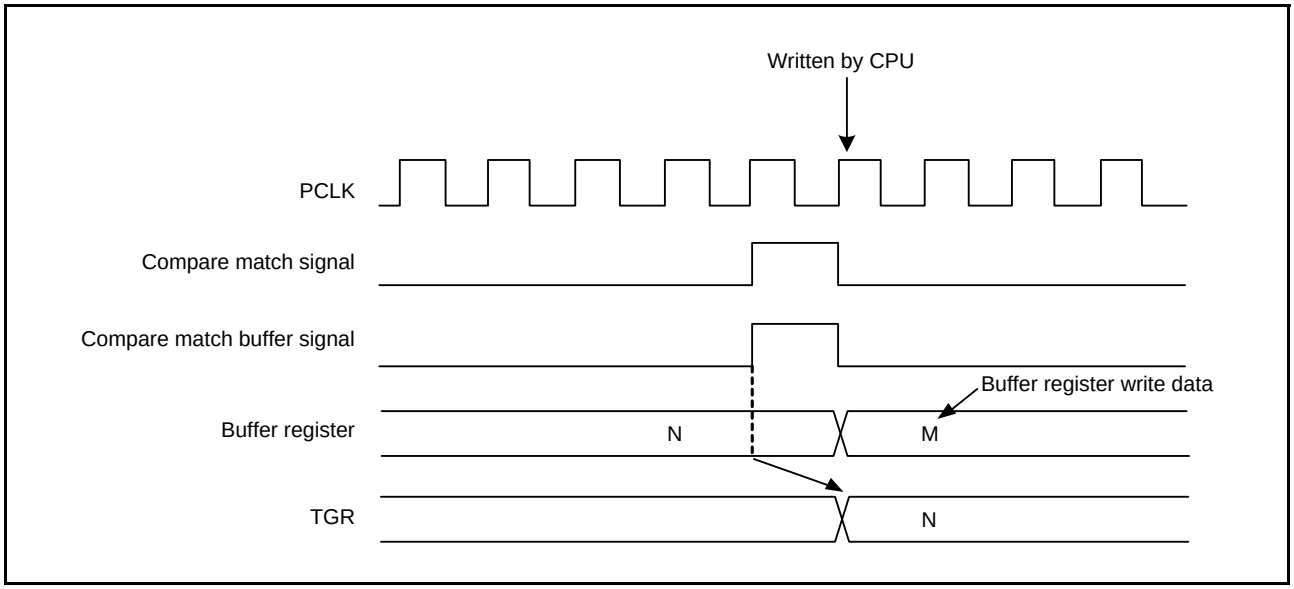



**Figure 20.110 Contention between Buffer Register Write Operation and Compare Match**

### 20.6.8 Contention between Buffer Register Write and TCNT Clear Operations

When the buffer transfer timing is set at the TCNT clear timing by the timer buffer operation transfer mode register (TBTM), if TCNT clearing occurs in a TGR write cycle, the data before write operation is transferred to TGR by the buffer operation.

Figure 20.111 shows the timing in this case.

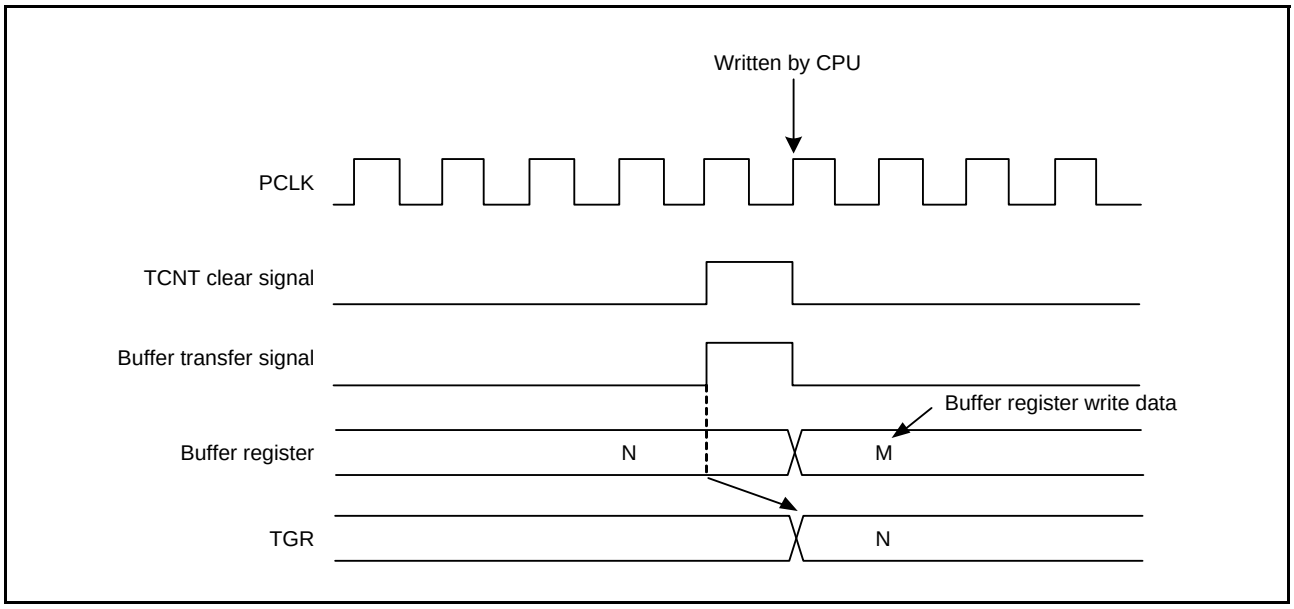



**Figure 20.111 Contention between Buffer Register Write and TCNT Clear Operations**

### 20.6.9 Contention between TGR Read Operation and Input Capture

If an input capture signal is generated in a TGR read cycle, the data before input capture transfer is read. Figure 20.112 shows the timing in this case.

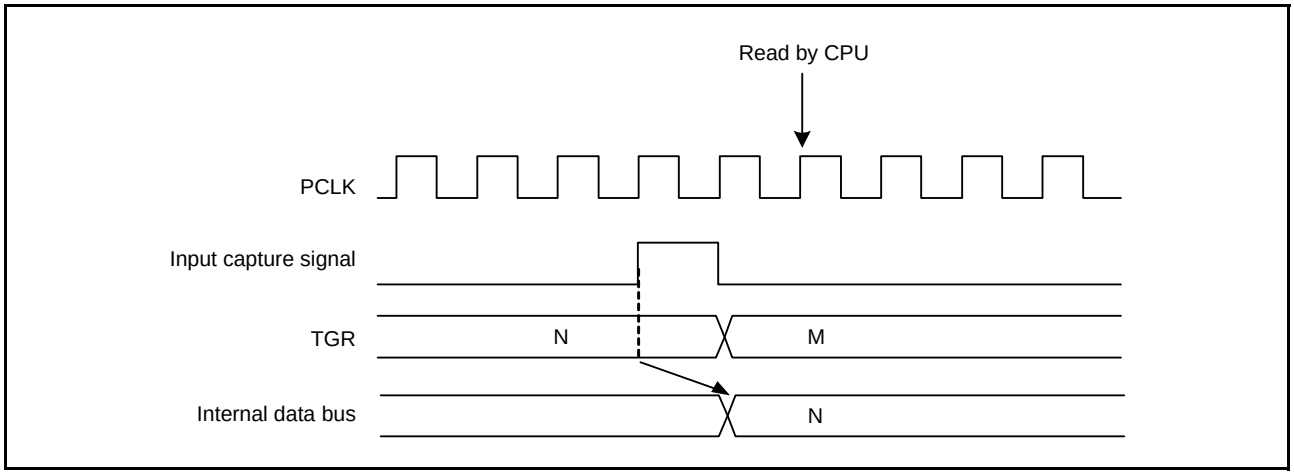



**Figure 20.112 Contention between TGR Read Operation and Input Capture (MTU0 to MTU5)**

### 20.6.10 Contention between TGR Write Operation and Input Capture

If an input capture signal is generated in a TGR write cycle, the input capture operation takes precedence and the TGR write operation is not performed in MTU0 to MTU4. In MTU5, the TGR write operation is performed and the input capture signal is generated.

Figure 20.113 and Figure 20.114 show the timing in this case.

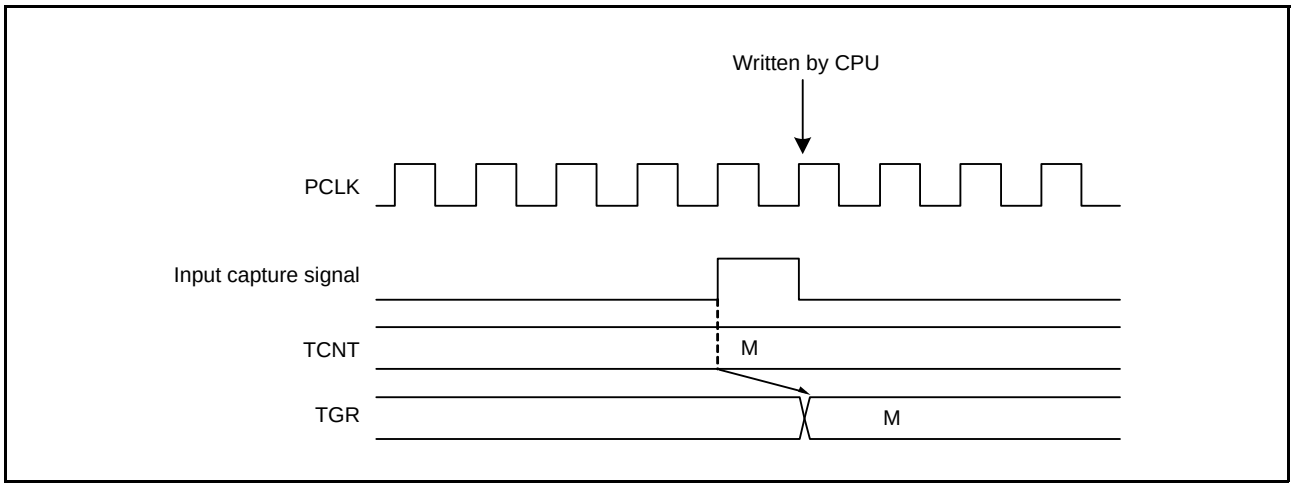



**Figure 20.113 Contention between TGR Write Operation and Input Capture (MTU0 to MTU4)**

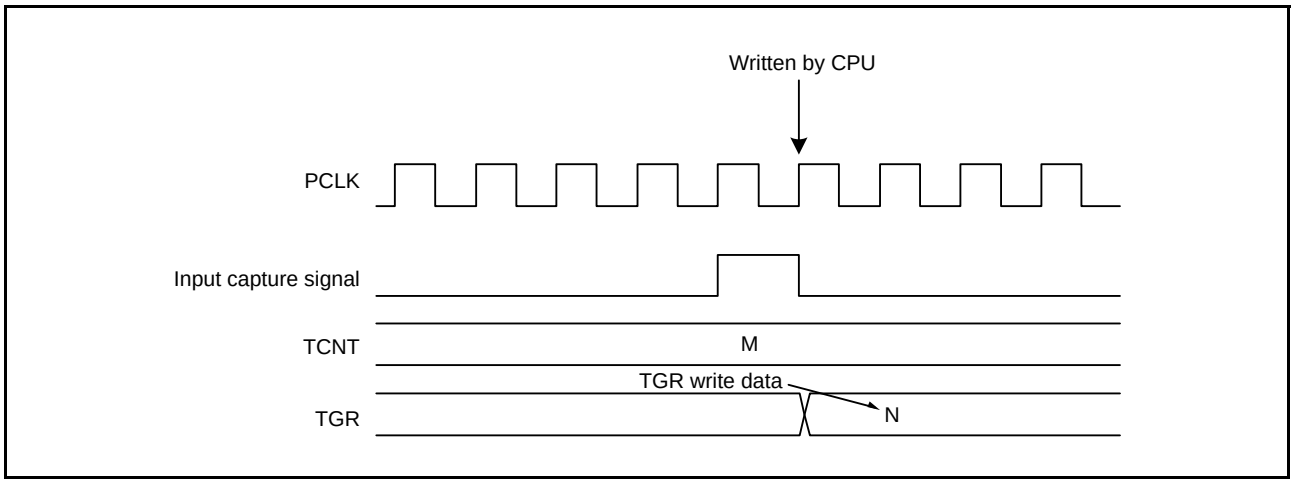



**Figure 20.114 Contention between TGR Write Operation and Input Capture (MTU5)**

### 20.6.11 Contention between Buffer Register Write Operation and Input Capture

If an input capture signal is generated in a buffer register write cycle, the buffer operation takes precedence and the buffer register write operation is not performed.

Figure 20.115 shows the timing in this case.

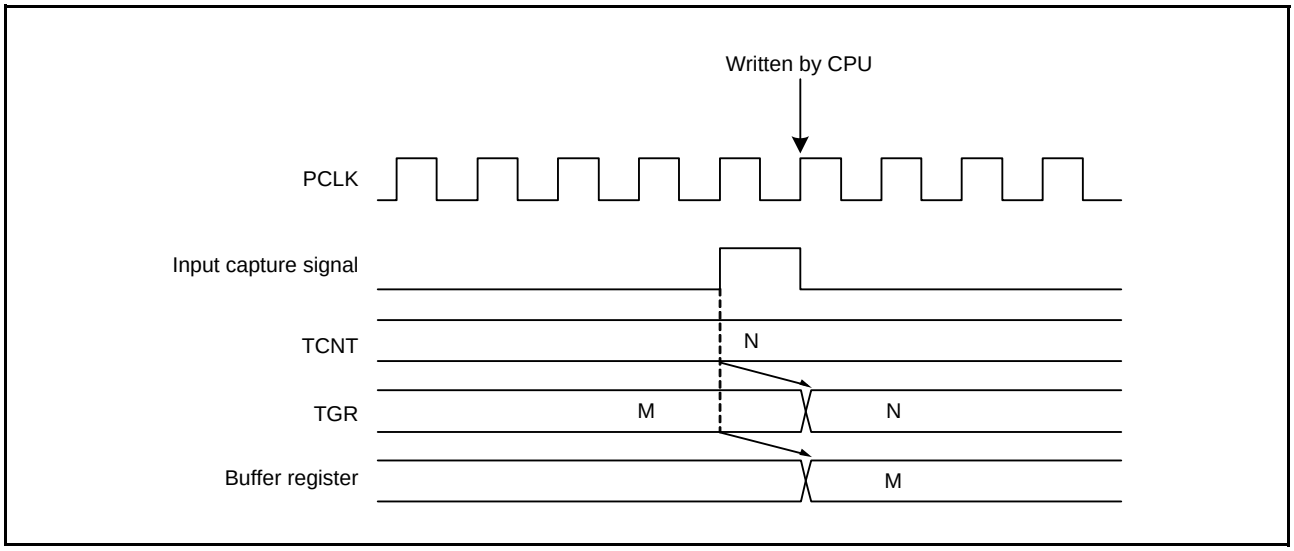



**Figure 20.115 Contention between Buffer Register Write Operation and Input Capture**

### 20.6.12 Contention between MTU2.TCNT Write Operation and Overflow/Underflow in Cascaded Operation

With timer counters MTU1.TCNT and MTU2.TCNT in a cascade, when a contention occurs between MTU1.TCNT counting (an MTU2.TCNT overflow/underflow) and the MTU2.TCNT write the MTU2.TCNT write operation is performed and the MTU1.TCNT count signal is disabled. In this case, if MTU1.TGRA works as a compare match register and there is a match between the MTU1.TGRA and MTU1.TCNT values, a compare match signal is issued. Furthermore, when the MTU1.TCNT count clock is selected as the input capture source of MTU0, MTU0.TGRA to MTU0.TGRD work in input capture mode. In addition, when the MTU0.TGRC compare match/input capture is selected as the input capture source of MTU1.TGRB, MTU1.TGRB works in input capture mode.
Figure 20.116 shows the timing in this case.
When setting the TCNT clearing function in cascaded operation, be sure to synchronize MTU1 and MTU2.

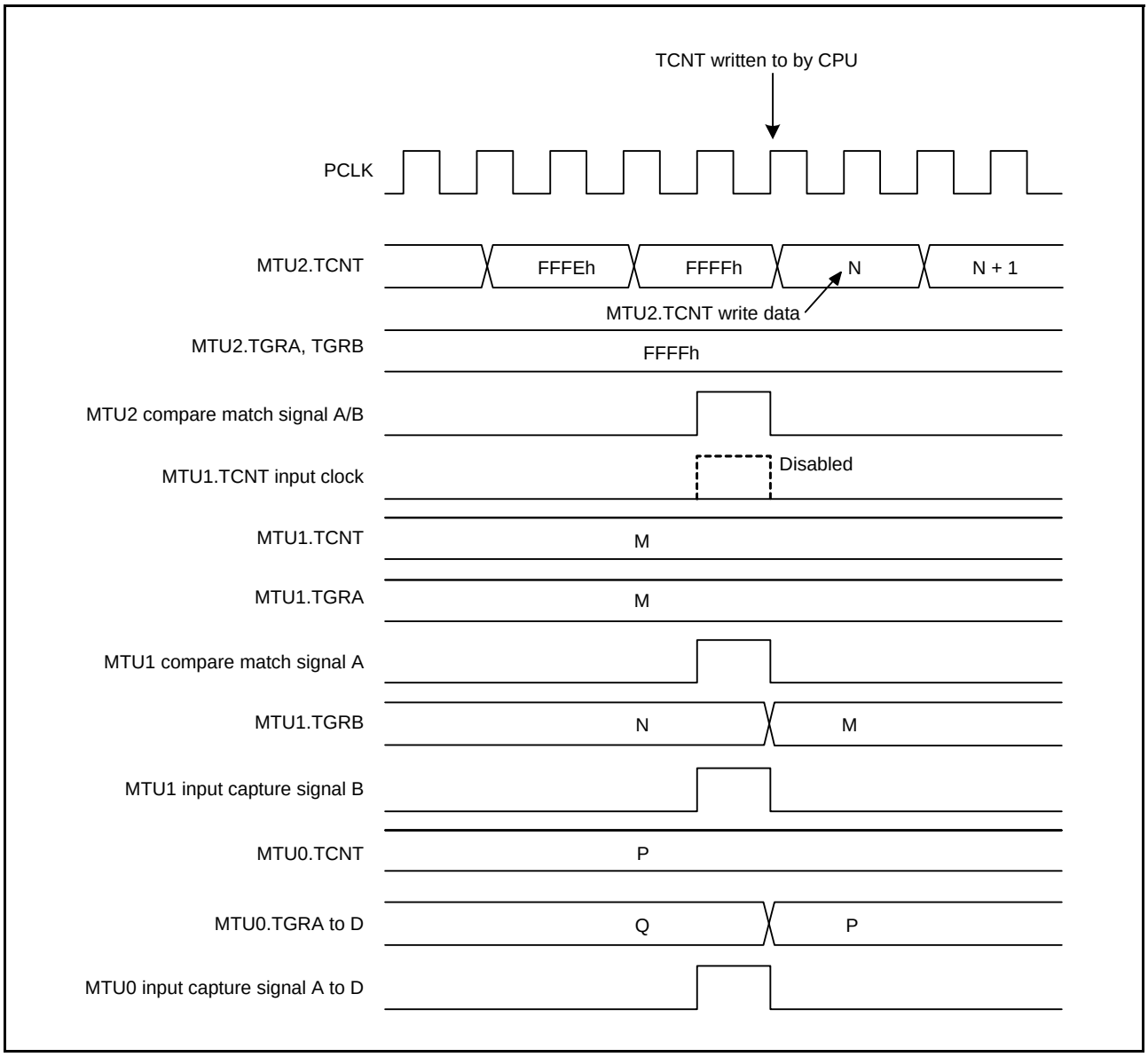



**Figure 20.116 Contention between MTU2.TCNT Write Operation and Overflow/Underflow in Cascaded Operation**

### 20.6.13 Counter Value when Stopped in Complementary PWM Mode

When counting operation in MTU3.TCNT and MTU4.TCNT is stopped in complementary PWM mode, MTU3.TCNT is set to the timer dead time register (TDDR) value and MTU4.TCNT is set to 0000h.
When operation is restarted in complementary PWM mode, counting begins automatically from the initial setting state. Figure 20.117 shows this operation.
When counting begins in another operating mode, be sure to make initial settings in MTU3.TCNT and MTU4.TCNT.

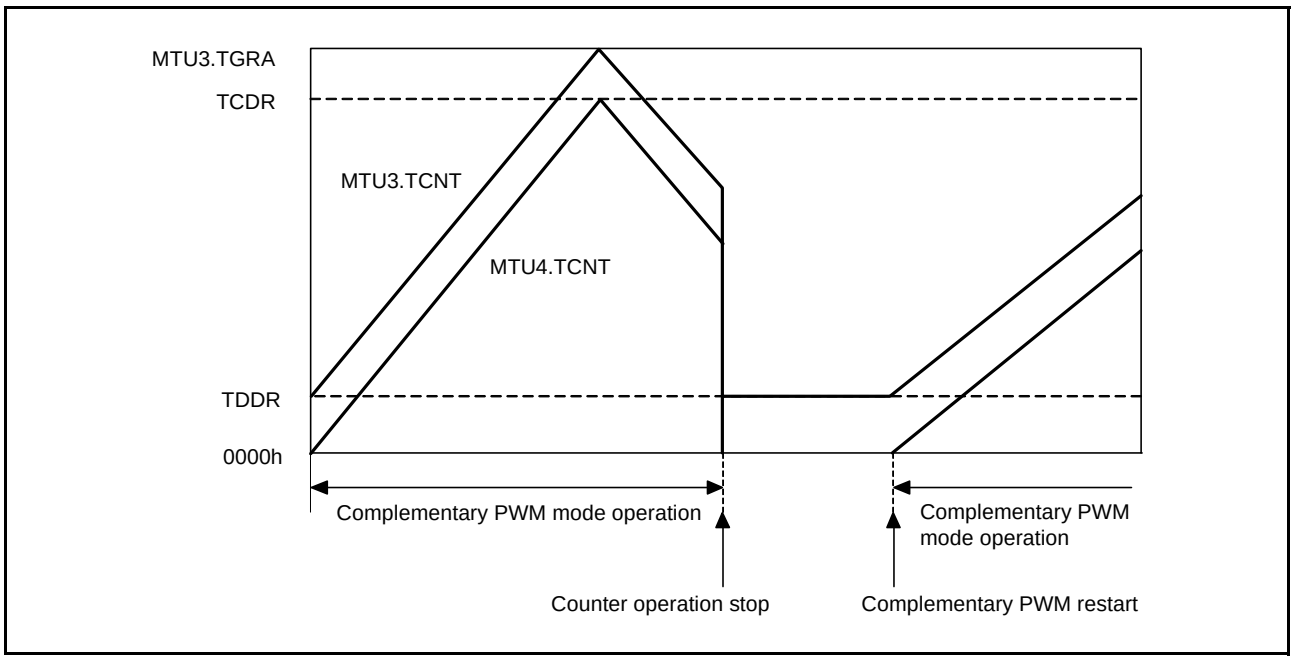



**Figure 20.117 Counter Value when Stopped in Complementary PWM Mode (MTU3 and MTU4 Operation)**

### 20.6.14 Buffer Operation Setting in Complementary PWM Mode

When modifying the PWM cycle set register (MTU3.TGRA), timer cycle data register (TCDR), and compare registers (MTU3.TGRB, MTU4.TGRA, and MTU4.TGRB) in complementary PWM mode, be sure to use buffer operation. Also, the MTU4.TMDR.BFA bit and MTU4.TMDR.BFB bit should be set to 0. Setting the BFA bit in MTU4.TMDR to 1 disables MTIOC4C pin waveform output. Setting the BFB bit in MTU4.TMDR to 1 also disables MTIOC4D pin waveform output.
In complementary PWM mode, buffer operation in MTU3 and MTU4 depends on the settings in bits BFA and BFB of MTU3.TMDR. When the BFA bit in MTU3.TMDR is set to 1, MTU3.TGRC functions as a buffer register for MTU3.TGRA. At the same time, MTU4.TGRC functions as a buffer register for MTU4.TGRA, and TCBR functions as a buffer register for TCDR.

### 20.6.15 Buffer Operation and Compare Match Flags in Reset-Synchronized PWM Mode

When setting buffer operation in reset-synchronized PWM mode, set the MTU4.TMDR.BFA bit and MTU4.TMDR.BFB bit to 0. Setting the MTU4.TMDR.BFA bit to 1 disables MTIOC4C pin waveform output. Setting the MTU4.TMDR.BFB bit to 1 also disables MTIOC4D pin waveform output.

In reset-synchronized PWM mode, buffer operation in MTU3 and MTU4 depends on the settings in the MTU3.TMDR.BFA bit and MTU3.TMDR.BFB bit. For example, if the MTU3.TMDR.BFA bit is set to 1, MTU3.TGRC functions as a buffer register for MTU3.TGRA. At the same time, MTU4.TGRC functions as a buffer register for MTU4.TGRA.

While the MTU3.TGRC and MTU3.TGRD are operating as buffer registers, the corresponding TGIC and TGID interrupt requests are never generated.

Figure 20.118 shows an example of MTU3.TGR, MTU4.TGR, MTIOC3m, and MTIOC4m operation with the MTU3.TMDR.BFA bit and MTU3.TMDR.BFB bit set to 1 and the MTU4.TMDR.BFA bit and MTU4.TMDR.BFB bit set to 0. (m = A to D)

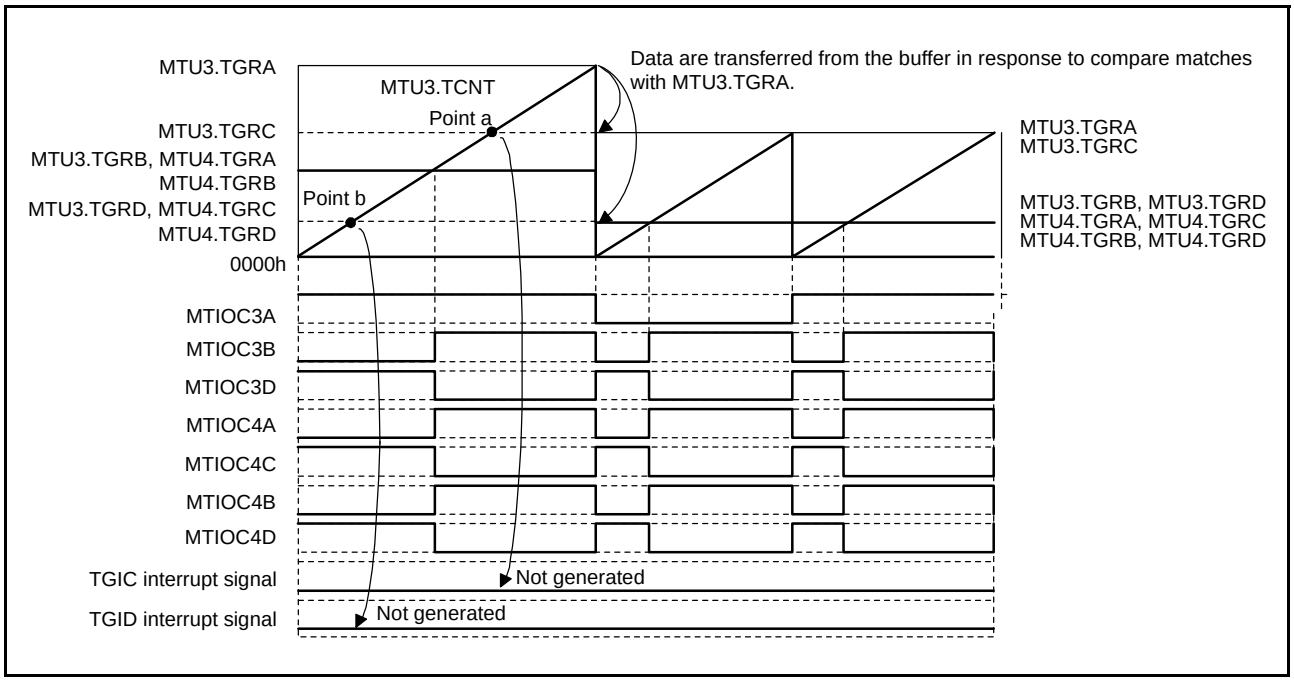



**Figure 20.118 Buffer Operation and Compare Match Flags in Reset-Synchronized PWM Mode**

### 20.6.16 Overflow Flags in Reset-Synchronized PWM Mode

After reset-synchronized PWM mode is selected, MTU3.TCNT and MTU4.TCNT start counting when the TSTR.CST3 bit is set to 1. In this state, the MTU4.TCNT count clock source and count edge are determined by the MTU3.TCR setting.

In reset-synchronized PWM mode, with cycle register MTU3.TGRA set to FFFFh and the MTU3.TGRA compare match selected as the counter clearing source, MTU3.TCNT and MTU4.TCNT count up to FFFFh, then a compare match occurs with MTU3.TGRA, and MTU3.TCNT and MTU4.TCNT are both cleared. In this case, the corresponding TCIV interrupt request is not generated.

Figure 20.119 shows an operation example in reset-synchronized PWM mode with cycle register MTU3.TGRA set to FFFFh and the MTU3.TGRA compare match specified for the counter clearing source.

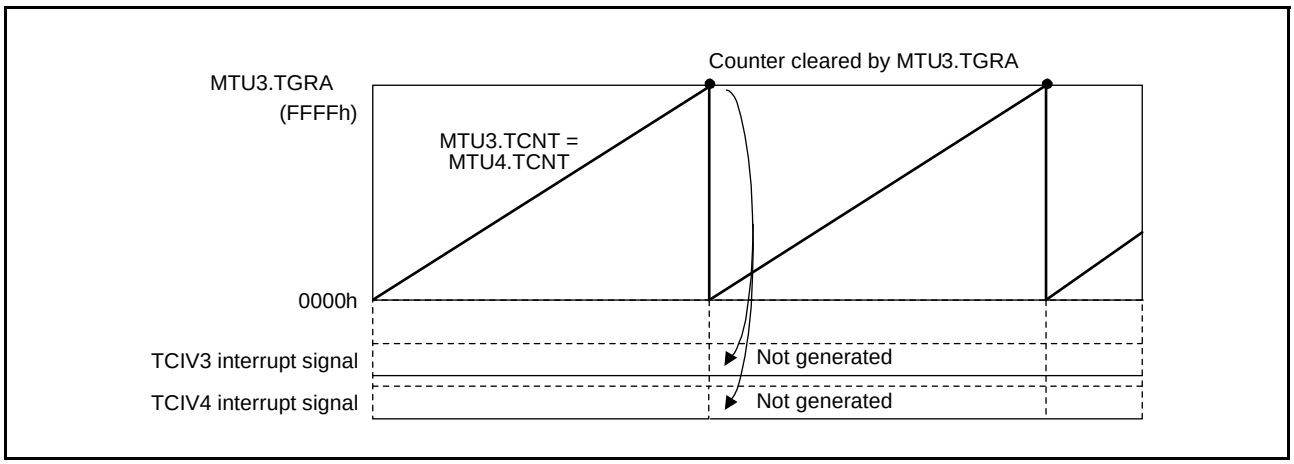



**Figure 20.119 Overflow Flags in Reset-Synchronized PWM Mode**

### 20.6.17 Contention between Overflow/Underflow and Counter Clearing

If an overflow/underflow and counter clearing occur simultaneously, TCNT clearing takes precedence and the corresponding TCIV interrupt is not generated. If an overflow and counter clearing due to an input capture occur simultaneously, an input capture interrupt signal is output and an overflow interrupt signal is not output.
Figure 20.120 shows the operation timing when a TGR compare match is specified as the clearing source and TGR is set to FFFFh.

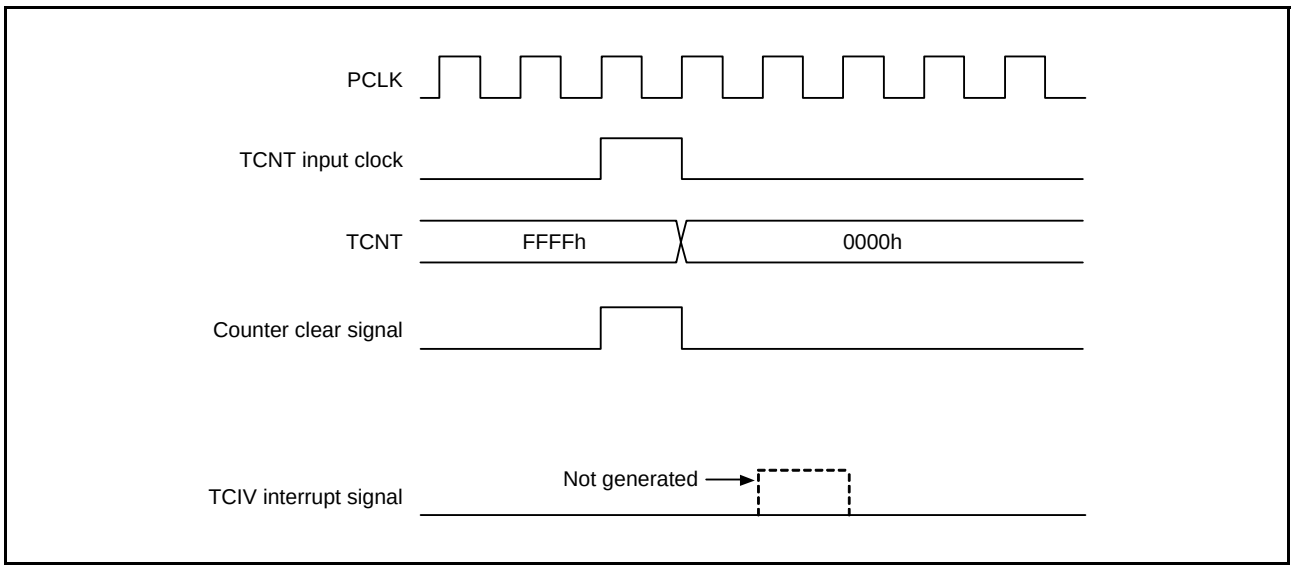



**Figure 20.120 Contention between Overflow and Counter Clearing**

### 20.6.18 Contention between TCNT Write Operation and Overflow/Underflow

If TCNT up-count or down-count in a TCNT write cycle and an overflow or an underflow occurs, the TCNT write operation takes precedence. The corresponding interrupt is not generated.
Figure 20.121 shows the operation timing when there is contention between TCNT write operation and overflow.

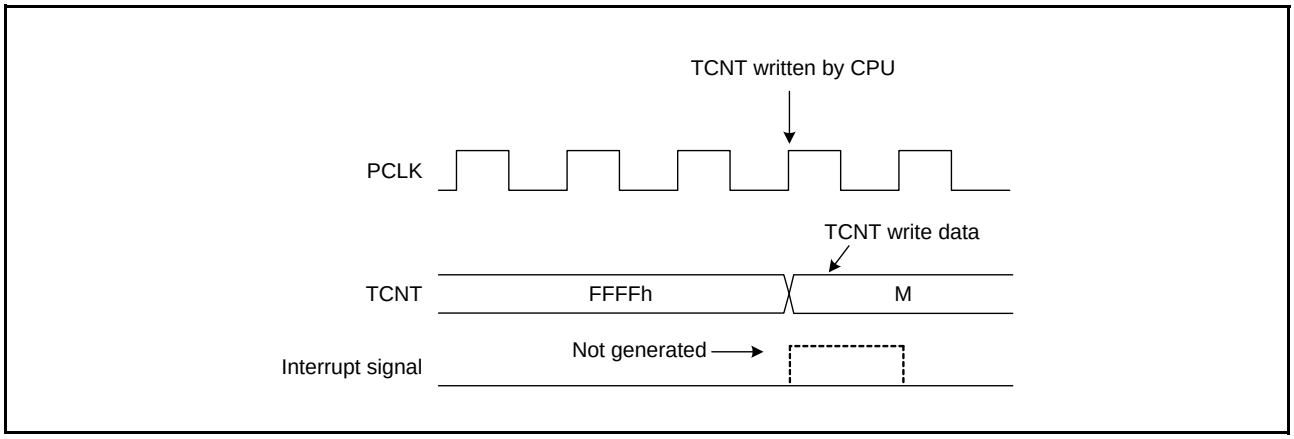



**Figure 20.121 Contention between TCNT Write Operation and Overflow**

### 20.6.19 Note on Transition from Normal Mode or PWM Mode 1 to Reset-Synchronized PWM Mode

When making a transition from normal mode or PWM mode 1 to reset-synchronized PWM mode in MTU3 and MTU4, if the counter is stopped while the output pins (MTIOC3B, MTIOC3D, MTIOC4A, MTIOC4C, MTIOC4B, and MTIOC4D) are held at a high level and then operation is started after a transition to reset-synchronized PWM mode, the initial pin output will not be correct.
When making a transition from normal mode to reset-synchronized PWM mode, write 11h to MTU3.TIORH, MTU3.TIORL, MTU4.TIORH, and MTU4.TIORL to initialize the output pin state to a low level, then set the registers to the initial value (00h) before making the mode transition.
When making a transition from PWM mode 1 to reset-synchronized PWM mode, switch to normal mode, initialize the output pin state to a low level, and then set the registers to the initial value (00h) before making the transition to reset-synchronized PWM mode.

### 20.6.20 Output Level in Complementary PWM Mode or Reset-Synchronized PWM Mode

When complementary PWM mode or reset-synchronized PWM mode is selected for MTU3 or MTU4, use the TOCR1.OLSP bit and TOCR1.OLSN bit to set the levels for PWM waveform output. Also, when either of these modes is in use, set TIOR to 00h. The negative-phase output level when the TDER.TDER bit is set to 0 (no dead time is generated) in complementary PWM mode is the inverse of the positive-phase output level according to the TOCR1.OLSP bit setting, not the TOCR1.OLSN bit setting.

### 20.6.21 Interrupts during Periods in the Module-Stop State

When an module that has issued an interrupt request enters the module-stop state, clearing the source of the interrupt for the CPU or activation signal for the DTC is not possible.
Accordingly, disable interrupts, etc. before making the settings for the module-stop state.

### 20.6.22 Simultaneous Input Capture in MTU1.TCNT and MTU2.TCNT in Cascade Connection

When timer counters (MTU1.TCNT and MTU2.TCNT) operate as a 32-bit counter in cascade connection, the cascaded counter value cannot be captured successfully in some cases even if input-capture input is simultaneously done to MTIOC1A and MTIOC2A or to MTIOC1B and MTIOC2B. This is because the input timing of MTIOC1A and MTIOC2A or of MTIOC1B and MTIOC2B may not be the same when external input-capture signals input into MTU1.TCNT and MTU2.TCNT are taken in synchronization with the internal clock.

For example, MTU1.TCNT (the counter for upper 16 bits) does not capture the count-up value by an overflow from MTU2.TCNT (the counter for lower 16 bits) but captures the count value before the up-counting. In this case, the values of MTU1.TCNT = FFF1h and MTU2.TCNT = 0000h should be transferred to MTU1.TGRA and MTU2.TGRA or to MTU1.TGRB and MTU2.TGRB, but the values of MTU1.TCNT = FFF0h and MTU2.TCNT = 0000h are erroneously transferred.

The MTU has a new function that allows simultaneous capture of MTU1.TCNT and MTU2.TCNT with a single input-capture input as the trigger. This function allows reading of the 32-bit counter such that MTU1.TCNT and MTU2.TCNT are captured at the same time. For details, refer to section 20.2.8, Timer Input Capture Control Register (TICCR).

### 20.6.23 Notes When Complementary PWM Mode Output Protection Functions are Not Used

The complementary PWM mode output protection functions are initially enabled. If the functions are not used, the POE.POECR2 register should be set to 00h.

### 20.6.24 Point for Caution Regarding MTU5.TCNT and MTU5.TGR Registers

Do not set an MTU5.TGRm (m = U, V, W) bit to the value of the corresponding MTU5.TCNTm (m = U, V, W) register plus one while counting by the MTU5.TCNTm (m = U, V, W) register is stopped. If an MTU5.TGRm (m = U, V, W) bit is set to the value of the corresponding MTU5.TCNTm (m = U, V, W) register plus one while counting by the MTU5.TCNTm (m = U, V, W) register is stopped, a compare-match will be generated even though counting is stopped. In this case, if the corresponding MTU5.TIER.TGIE5m (m = U, V, W) bit is also set to 1 (interrupt requests enabled), a compare-match interrupt will also be generated. If the value of the timer compare match clear register is also 1 (enabled), the timer is automatically cleared to 0000h when the compare-match is generated, regardless of whether interrupts from the MTU5.TCNTm (m = U, V, W) are enabled or disabled.

### 20.6.25 Points for Caution to Prevent Malfunctions in Synchronous Clearing for Complementary PWM Mode

If control of the output waveform is enabled (TWCR.WRE = 1) at the time of synchronous counter clearing in complementary PWM mode, satisfaction of either condition 1 or 2 below has the following effects.

- Dead time on the PWM output pins is shortened (or disappears).
- The active level is output on the PWM inverse-phase output pins beyond the period for active-level output.

Condition 1: In portion (10) of the initial output inhibition period in Figure 20.122, synchronous clearing occurs within the dead-time period for PWM output.

Condition 2: In portions (10) and (11) of the initial output inhibition period in Figure 20.123, synchronous clearing occurs when any condition from among MTU3.TGRB ≤ TDDR, MTU4.TGRA ≤ TDDR, or MTU4.TGRB ≤TDDR is satisfied.

The following method avoids the above phenomena.

- Ensure that synchronous clearing proceeds with the value of each comparison register (MTU3.TGRB, MTU4.TGRA, and MTU4.TGRB) set to at least double the value of the dead time data register (TDDR).

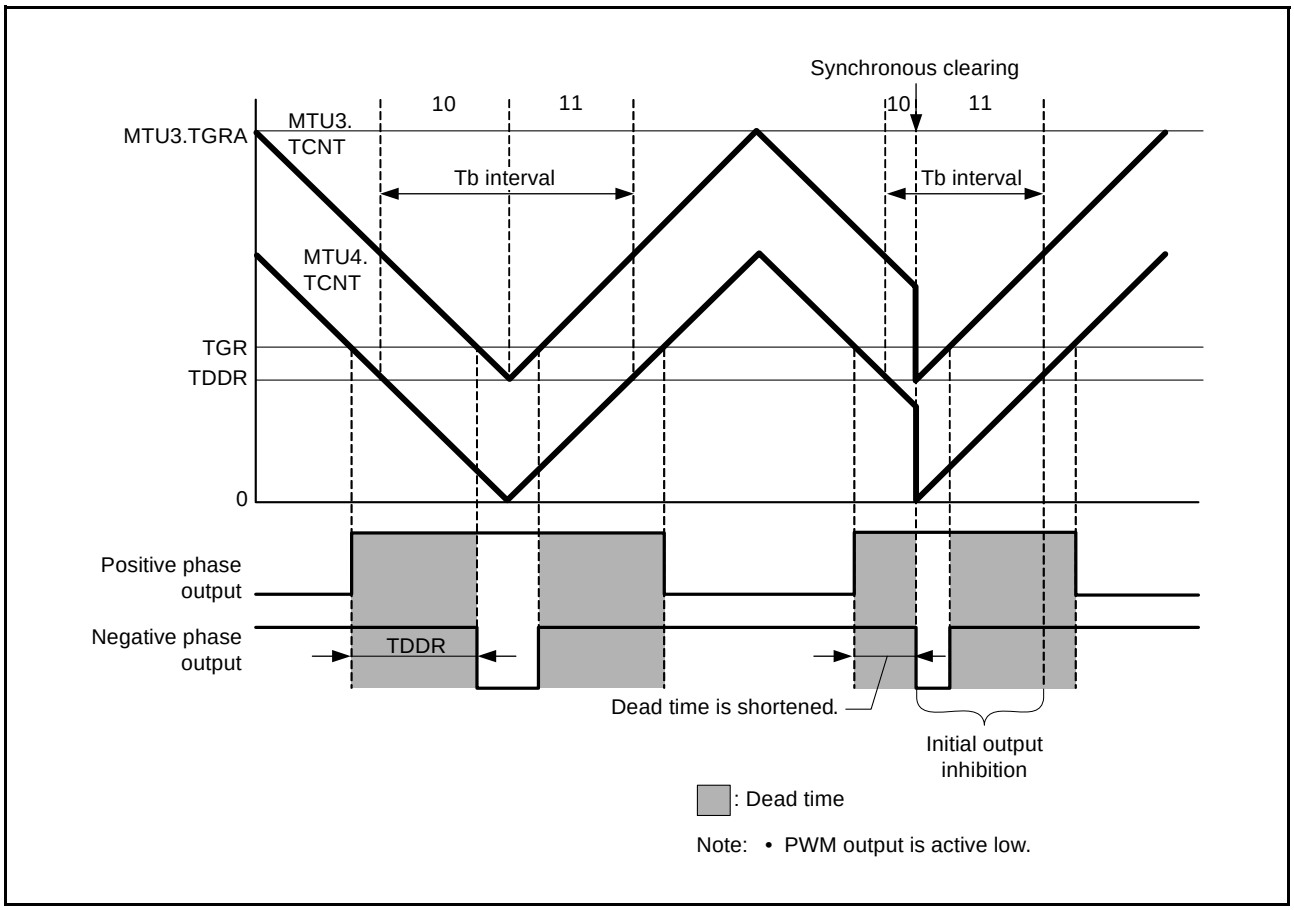



**Figure 20.122 Example of Synchronous Clearing (When Condition 1 Applies)**

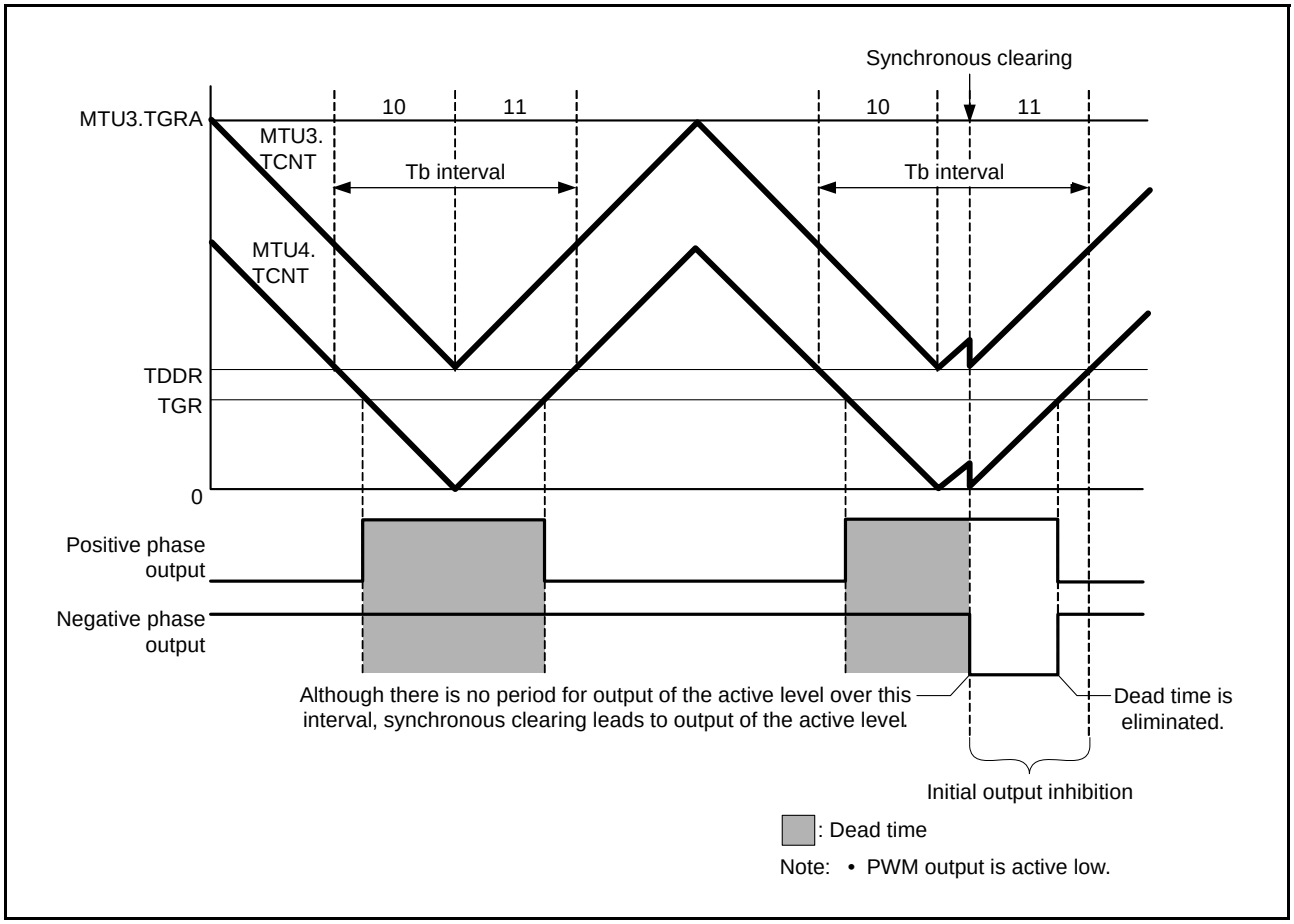



**Figure 20.123 Example of Synchronous Clearing (When Condition 2 Applies)**

### 20.6.26 Continuous Output of Interrupt Signal in Response to a Compare Match

When TGR is set to 0000h, PCLK/1 clock is set as the counter clock, and compare match is set as the trigger for clearing of the counter clock, the value of the counter (TCNT) counter remains 0000h, and the interrupt signal will be output continuously (i.e. its level will be flat) rather than output over a single cycle. Consequently, interrupts will not be detected in response to second and subsequent compare matches.

Figure 20.124 shows the timing for continuous output of the interrupt signal in response to a compare match.

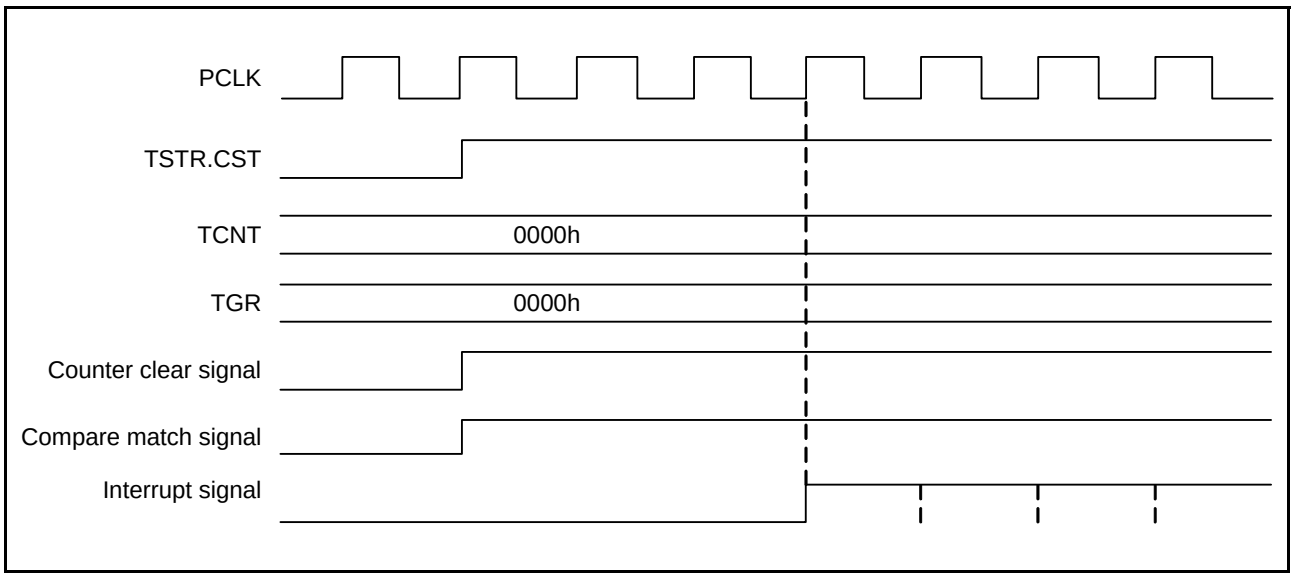



**Figure 20.124 Continuous Output of Interrupt Signal in Response to a Compare Match**

## 20.7 MTU Output Pin Initialization

### 20.7.1 Operating Modes

The MTU has the following six operating modes. Waveforms can be output in any of these modes.

- Normal mode (MTU0 to MTU4)
- PWM mode 1 (MTU0 to MTU4)
- PWM mode 2 (MTU0 to MTU2)
- Phase counting modes 1 to 4 (MTU1 and MTU2)
- Complementary PWM mode (MTU3 and MTU4)
- Reset-synchronized PWM mode (MTU3 and MTU4)

This section describes how to initialize the MTU output pins in each of these modes.

### 20.7.2 Operation in Case of Re-Setting Due to Error during Operation

If an error occurs during MTU operation, MTU output should be cut off by the system. For an I/O port that is shut down, set the port-direction registers (PDR), the port output data register (PODR), and the port mode register (PMR) to switch the port pins to be general output pins and for output of the non-active level. Set the TIOR for the MTU pins to disable output. Set the TOER for the complementary PWM output pins (MTIOC3B, MTIOC3D, MTIOC4A, MTIOC4B, MTIOC4C, and MTIOC4D). For PWM output pins, output can also be cut by hardware, using port output enable 2(POE). The pin initialization procedures for re-setting due to an error during operation and the procedures for restarting in a different mode after re-setting are described below.
The MTU has six operating modes, as stated above. There are thus 36 mode transition combinations, but some transitions are not available with certain channel and mode combinations. Available mode transition combinations are listed in Table 20.59.

Note that the following notations are used for operating modes.

Normal: Normal mode PWM1: PWM mode 1 PWM2: PWM mode 2
PCM: Phase counting modes 1 to 4 CPWM: Complementary PWM mode RPWM: Reset-synchronized PWM mode

**Table 20.59 Mode Transition Combinations**

| | Normal | PWM1 | PWM2 | PCM | CPWM | RPWM |
|---|---|---|---|---|---|---|
| Normal | (1) | (2) | (3) | (4) | (5) | (6) |
| PWM1 | (7) | (8) | (9) | (10) | (11) | (12) |
| PWM2 | (13) | (14) | (15) | (16) | Not available | Not available |
| PCM | (17) | (18) | (19) | (20) | Not available | Not available |
| CPWM | (21) | (22) | Not available | Not available | (23)(24) | (25) |
| RPWM | (26) | (27) | Not available | Not available | (28) | (29) |

### 20.7.3 Overview of Pin Initialization Procedures and Mode Transitions in Case of Error during Operation

- When making a transition to a mode (Normal, PWM1, PWM2, or PCM) in which the pin output level is selected by the timer I/O control register (TIOR) setting, initialize the pins by means of TIOR setting.
- In PWM mode 1, a waveform is not output to the MTIOCnB and MTIOCnD (n = 3, 4) pins. If there is no other module that outputs to the corresponding pins, the pins enter high-impedance state. If there is a level to be output, make general output port settings using the port direction register (PDR), port output data register (PODR), and port mode register (PMR) for I/O ports.
- In PWM mode 2, a waveform is not output to the cycle register pins. If there is no other module that outputs to the corresponding pins, the pins enter high-impedance state. If there is a level to be output, make general output port settings using the port direction register (PDR), port output data register (PODR), and port mode register (PMR) for I/O ports.
- In normal mode or PWM 2 mode, if the TGRC and TGRD operate as buffer registers, a waveform is not output to the pins. If there is no other module that outputs to the corresponding pins, the pins enter high-impedance state. If there is a level to be output, make general output port settings using the port direction register (PDR), port output data register (PODR), and port mode register (PMR) for I/O ports.
- In PWM mode 1, if either TGRC or TGRD operates as a buffer register and there is no other module that outputs to the corresponding pins, the pins enter high-impedance state. If there is a level to be output, make general output port settings using the port direction register (PDR), port output data register (PODR), and port mode register (PMR) for I/O ports.
- When making a transition to a mode (CPWM or RPWM) in which the pin output level is selected by the timer output control register (TOCR) setting, temporarily disable output in MTU3 and MTU4 with the timer output master enable register (TOER). If there is no other module that outputs to the corresponding pins, the pins enter high-impedance state. If there is a level to be output, make general output port settings using the port direction register (PDR), port output data register (PODR), and port mode register (PMR) for I/O ports. Switch to normal mode, perform initialization with the TIOR register, and restore the TIOR register to its initial value. After that, operate the MTU in accordance with the mode setting procedure (TOCR setting, TMDR setting, and TOER setting).

Note: • Channel number is substituted for “n” indicated in this section unless otherwise specified.

Pin initialization procedures are described below for the numbered combinations in Table 20.59. The active level is assumed to be low.

### (1) Operation When Error Occurs in Normal Mode and Operation is Restarted in Normal Mode

Figure 20.125 shows a case in which an error occurs in normal mode and operation is restarted in normal mode after re-setting.

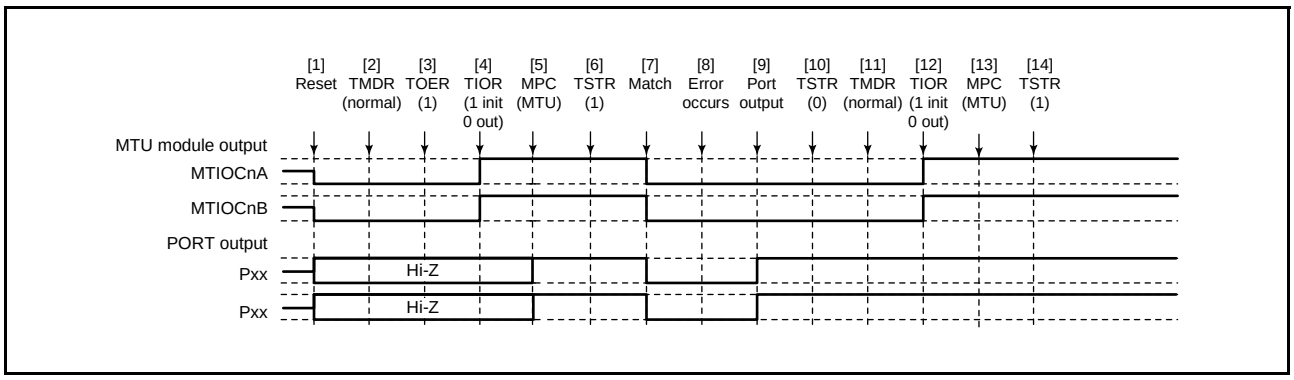



**Figure 20.125 Error Occurrence in Normal Mode, Recovery in Normal Mode**

[1] After a reset, the MTU output goes low and the ports enter high-impedance state.
[2] After a reset, the TMDR setting is for normal mode.
[3] For MTU3 and MTU4, enable output with TOER before initializing the pins with TIOR.
[4] Initialize the pins with TIOR. (In the example, the initial output is a high level, and a low level is output on compare match occurrence.)
[5] Use the MPC and the port mode register (PMR) for the I/O port to set up MTU output.
[6] Start count operation by setting TSTR.
[7] Output goes low on compare match occurrence.
[8] An error occurs.
[9] Use the port direction register (PDR) and port mode register (PMR) for the input port pin to switch it to operate as a general output port pin, and the port output data register (PODR) to select output of the non-active level.
[10] Stop count operation by setting TSTR.
[11] This step is not necessary when restarting in normal mode.
[12] Initialize the pins with TIOR.
[13] Use the MPC and the port mode register (PMR) for the I/O port to set up MTU output.
[14] Restart operation by setting TSTR.

### (2) Operation When Error Occurs in Normal Mode and Operation is Restarted in PWM Mode 1

Figure 20.126 shows a case in which an error occurs in normal mode and operation is restarted in PWM mode 1 after re-setting.

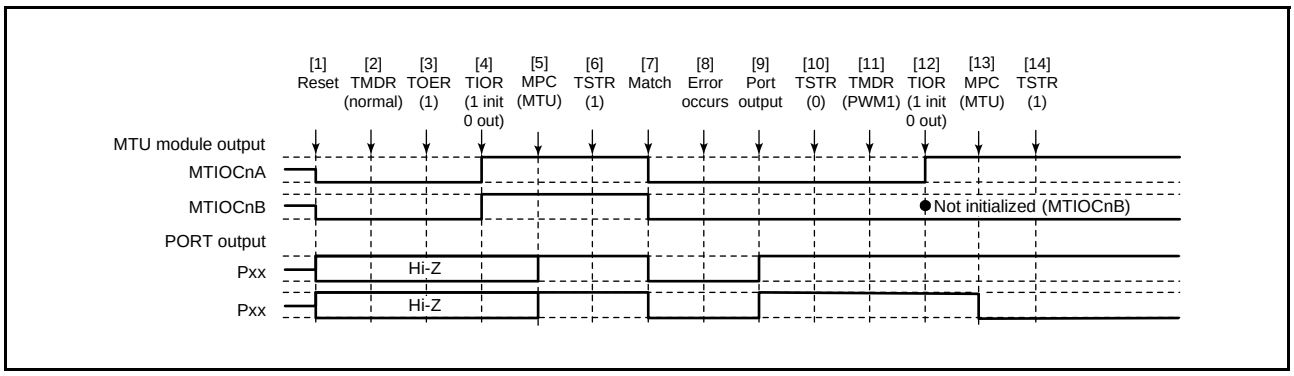



**Figure 20.126 Error Occurrence in Normal Mode, Recovery in PWM Mode 1**

[1] to [10] are the same as in Figure 20.125.

[11] Set PWM mode 1.

[12] Set the TIOR register to initialize pins, i.e. so that the MTIOCnB (or MTIOCnD) does not produce a waveform in PWM mode 1. If a particular level should be output, set the port direction register (PDR) and the port output data register (PODR) so that the pins of the I/O port operate as general outputs.

[13] Use the MPC and the port mode register (PMR) for the I/O port to set up MTU output.

[14] Restart operation by setting TSTR.

### (3) Operation When Error Occurs in Normal Mode and Operation is Restarted in PWM Mode 2

Figure 20.127 shows a case in which an error occurs in normal mode and operation is restarted in PWM mode 2 after re-setting.

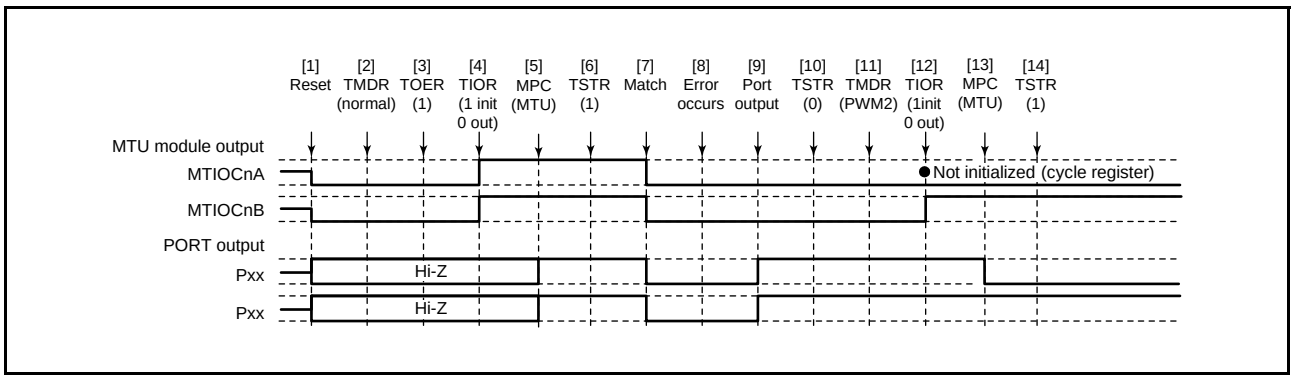



**Figure 20.127 Error Occurrence in Normal Mode, Recovery in PWM Mode 2**

[1] to [10] are the same as in Figure 20.125.

[11] Set PWM mode 2.

[12] Initialize the pins with TIOR. (In PWM mode 2, the cycle register pins are not initialized. If initialization is required, initialize in normal mode, then switch to PWM mode 2.)

[13] Use the MPC and the port mode register (PMR) for the I/O port to set up MTU output.

[14] Restart operation by setting TSTR.

Note: • PWM mode 2 can only be selected for MTU0 to MTU2, and therefore TOER setting is not necessary.

### (4) Operation When Error Occurs in Normal Mode and Operation is Restarted in Phase Counting Mode

Figure 20.128 shows a case in which an error occurs in normal mode and operation is restarted in phase counting mode after re-setting.

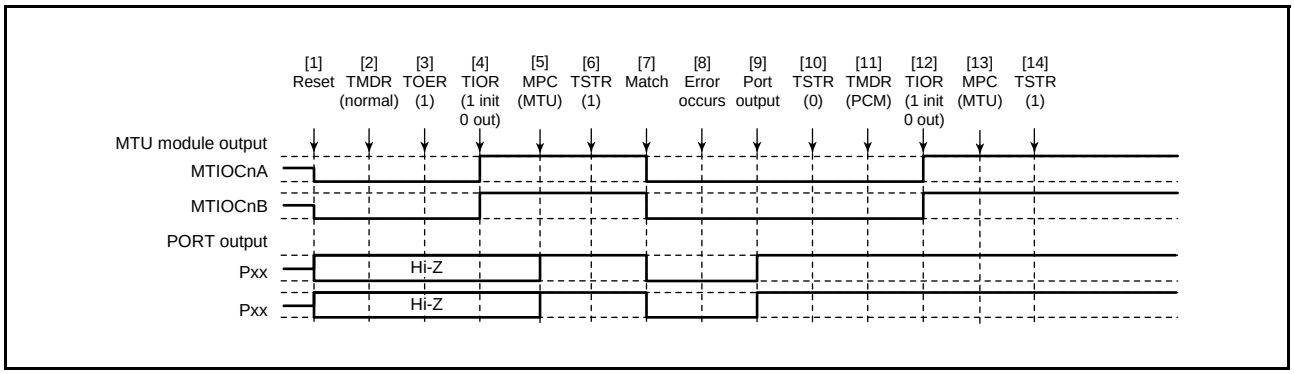



**Figure 20.128 Error Occurrence in Normal Mode, Recovery in Phase Counting Mode**

[1] to [10] are the same as in Figure 20.125.

[11] Set the phase counting mode.

[12] Initialize the pins with TIOR.

[13] Use the MPC and the port mode register (PMR) for the I/O port to set up MTU output.

[14] Restart operation by setting TSTR.

Note: • The phase counting mode can only be selected for MTU1 and MTU2, and therefore TOER setting is not necessary.

### (5) Operation When Error Occurs in Normal Mode and Operation is Restarted in Complementary PWM Mode

Figure 20.129 shows a case in which an error occurs in normal mode and operation is restarted in complementary PWM mode after re-setting.

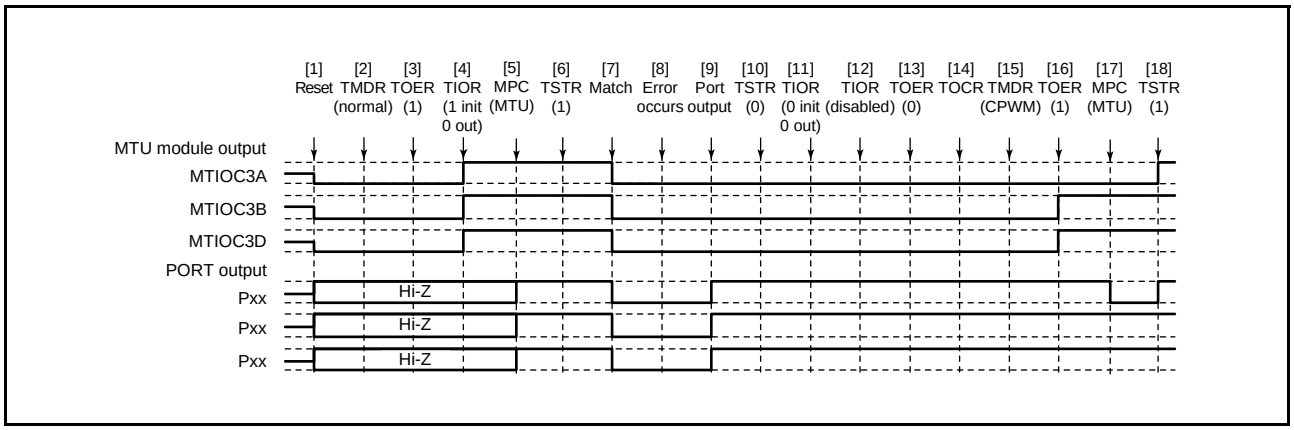



**Figure 20.129 Error Occurrence in Normal Mode, Recovery in Complementary PWM Mode**

[1] to [10] are the same as in Figure 20.125.
[11] Initialize the normal mode waveform generation section with TIOR.
[12] Disable operation of the normal mode waveform generation section with TIOR.
[13] Disable output in MTU3 and MTU4 with TOER.
[14] Select the complementary PWM output level and enable or disable cyclic output with TOCR.
[15] Set complementary PWM mode.
[16] Enable output in MTU3 and MTU4 with TOER.
[17] Use the MPC and the port mode register (PMR) for the I/O port to set up MTU output.
[18] Restart operation by setting TSTR.

### (6) Operation When Error Occurs in Normal Mode and Operation is Restarted in Reset-Synchronized PWM Mode

Figure 20.130 shows a case in which an error occurs in normal mode and operation is restarted in reset-synchronized PWM mode after re-setting.

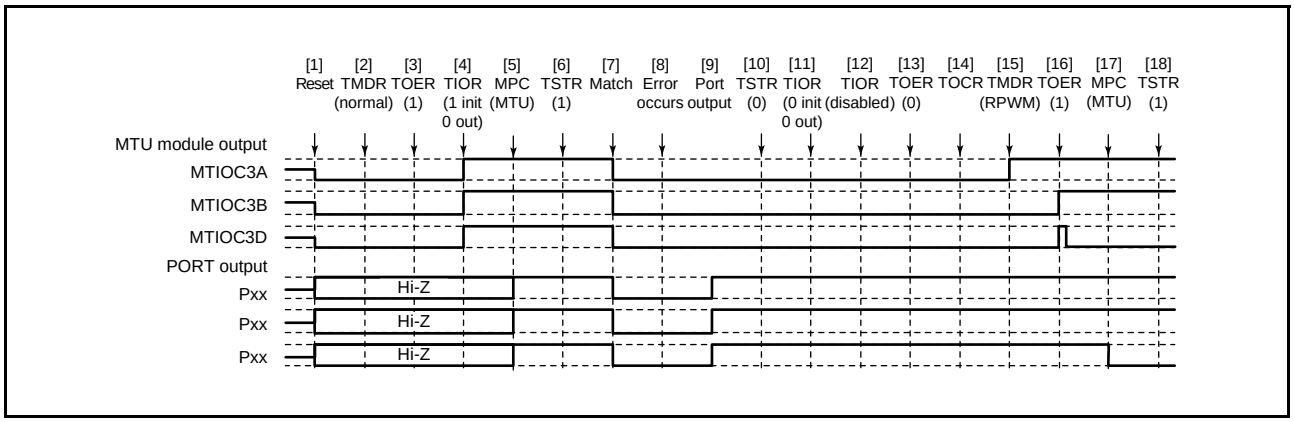



**Figure 20.130 Error Occurrence in Normal Mode, Recovery in Reset-Synchronized PWM Mode**

[1] to [13] are the same as in Figure 20.125.

[14] Select the reset-synchronized PWM output level and enable or disable cyclic output with TOCR.

[15] Set reset-synchronized PWM mode.

[16] Enable output in MTU3 and MTU4 with TOER.

[17] Use the MPC and the port mode register (PMR) for the I/O port to set up MTU output.

[18] Restart operation by setting TSTR.

### (7) Operation When Error Occurs in PWM Mode 1 and Operation is Restarted in Normal Mode

Figure 20.131 shows a case in which an error occurs in PWM mode 1 and operation is restarted in normal mode after re-setting.

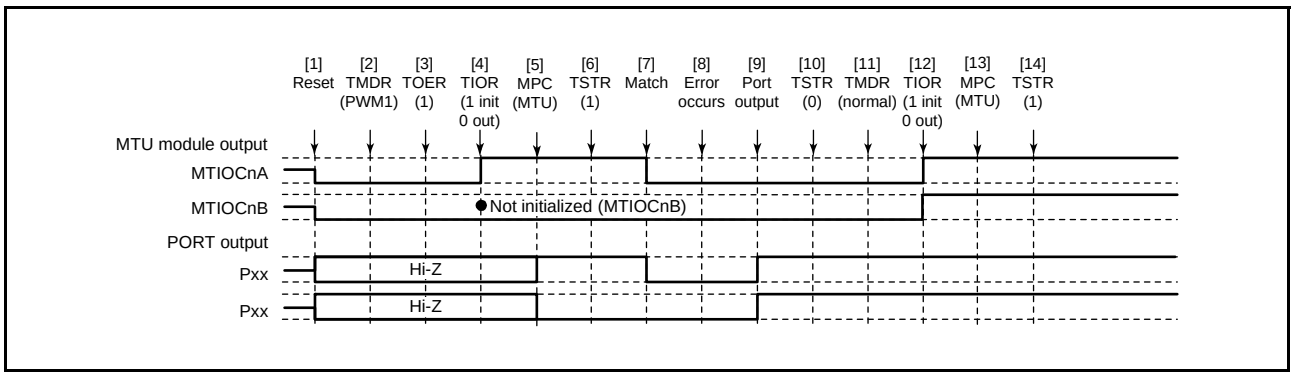



**Figure 20.131 Error Occurrence in PWM Mode 1, Recovery in Normal Mode**

[1] After a reset, the MTU output goes low and the ports enter high-impedance state.
[2] Set PWM mode 1.
[3] For MTU3 and MTU4, enable output with TOER before initializing the pins with TIOR.
[4] Initialize the pins with TIOR. (In the example, the initial output is a high level, and a low level is output on compare match occurrence. In PWM mode 1, the MTIOCnB side is not initialized.)
[5] Use the MPC and the port mode register (PMR) for the I/O port to set up MTU output.
[6] Start count operation by setting TSTR.
[7] Output goes low on compare match occurrence.
[8] An error occurs.
[9] Use the port direction register (PDR) and port mode register (PMR) for the input port pin to switch it to operate as a general output port pin, and the port output data register (PODR) to select output of the non-active level.
[10] Stop count operation by setting TSTR.
[11] Set normal mode.
[12] Initialize the pins with TIOR.
[13] Use the MPC and the port mode register (PMR) for the I/O port to set up MTU output.
[14] Restart operation by setting TSTR.

### (8) Operation When Error Occurs in PWM Mode 1 and Operation is Restarted in PWM mode 1

Figure 20.132 shows a case in which an error occurs in PWM mode 1 and operation is restarted in PWM mode 1 after re-setting.

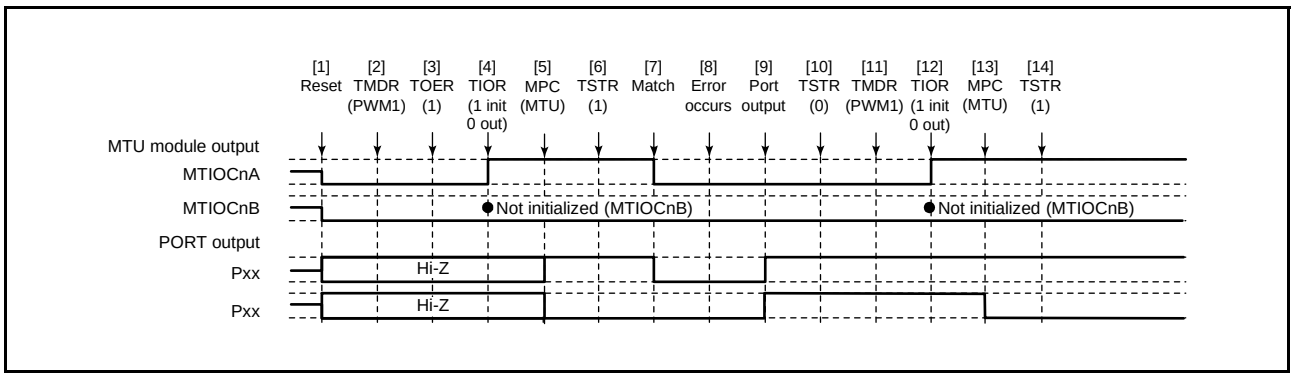



**Figure 20.132 Error Occurrence in PWM Mode 1, Recovery in PWM Mode 1**

[1] to [10] are the same as in Figure 20.131.

[11] This step is not necessary when restarting in PWM mode 1.

[12] Initialize the pins with the TIOR register. (In PWM mode 1, a waveform is not output on the MTIOCnB (MTIOCnD) pins. If a level is to be output, make the required general output port settings in the I/O port's port direction register (PDR) and port output data register (PODR).)

[13] Use the MPC and the port mode register (PMR) for the I/O port to set up MTU output.

[14] Restart operation by setting TSTR.

### (9) Operation When Error Occurs in PWM Mode 1 and Operation is Restarted in PWM mode 2

Figure 20.133 shows a case in which an error occurs in PWM mode 1 and operation is restarted in PWM mode 2 after re-setting.

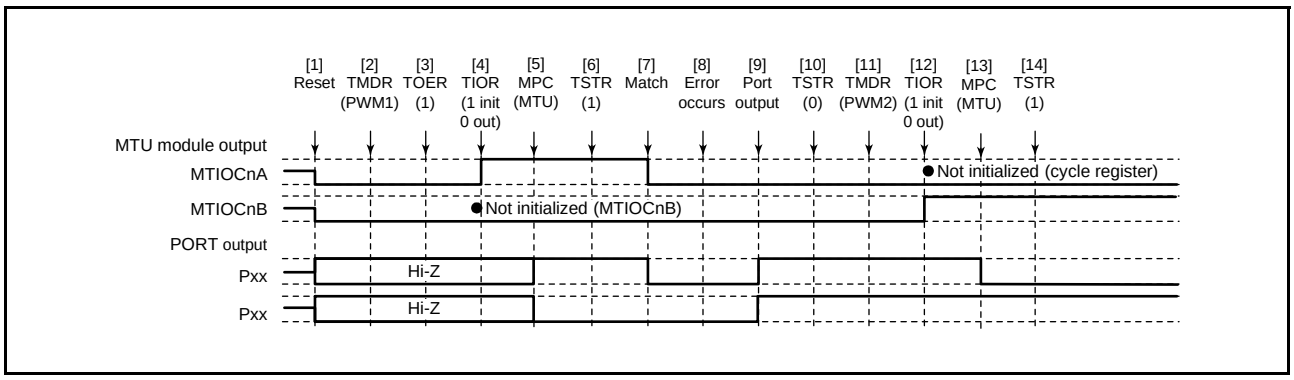



**Figure 20.133 Error Occurrence in PWM Mode 1, Recovery in PWM Mode 2**

[1] to [10] are the same as in Figure 20.131.

[11] Set PWM mode 2.

[12] Initialize the pins with the TIOR register. (In PWM mode 2, a waveform is not output on the cycle register pins. If a level is to be output, make the required general output port settings in the I/O port's port direction register (PDR) and port output data register (PODR).)

[13] Use the MPC and the port mode register (PMR) for the I/O port to set up MTU output.

[14] Restart operation by setting TSTR.

Note: • PWM mode 2 can only be selected for MTU0 to MTU2, and therefore, TOER setting is not necessary.

### (10) Operation When Error Occurs in PWM Mode 1 and Operation is Restarted in Phase Counting Mode

Figure 20.134 shows a case in which an error occurs in PWM mode 1 and operation is restarted in phase counting mode after re-setting.

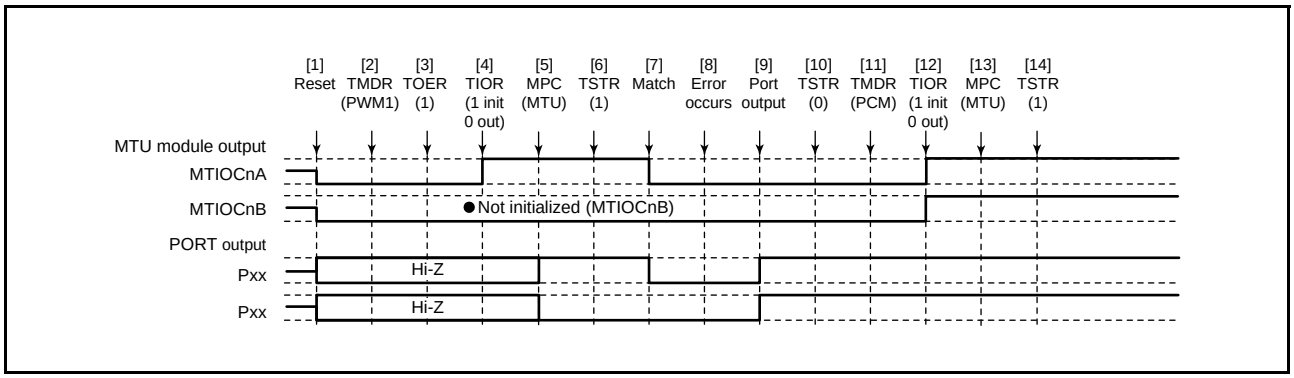



**Figure 20.134 Error Occurrence in PWM Mode 1, Recovery in Phase Counting Mode**

[1] to [10] are the same as in Figure 20.131.

[11] Set the phase counting mode.

[12] Initialize the pins with TIOR.

[13] Use the MPC and the port mode register (PMR) for the I/O port to set up MTU output.

[14] Restart operation by setting TSTR.

Note: • The phase counting mode can only be selected for MTU1 and MTU2, and therefore TOER setting is not necessary.

## (11) Operation When Error Occurs in PWM Mode 1 and Operation is Restarted in Complementary PWM Mode

Figure 20.135 shows a case in which an error occurs in PWM mode 1 and operation is restarted in complementary PWM mode after re-setting.

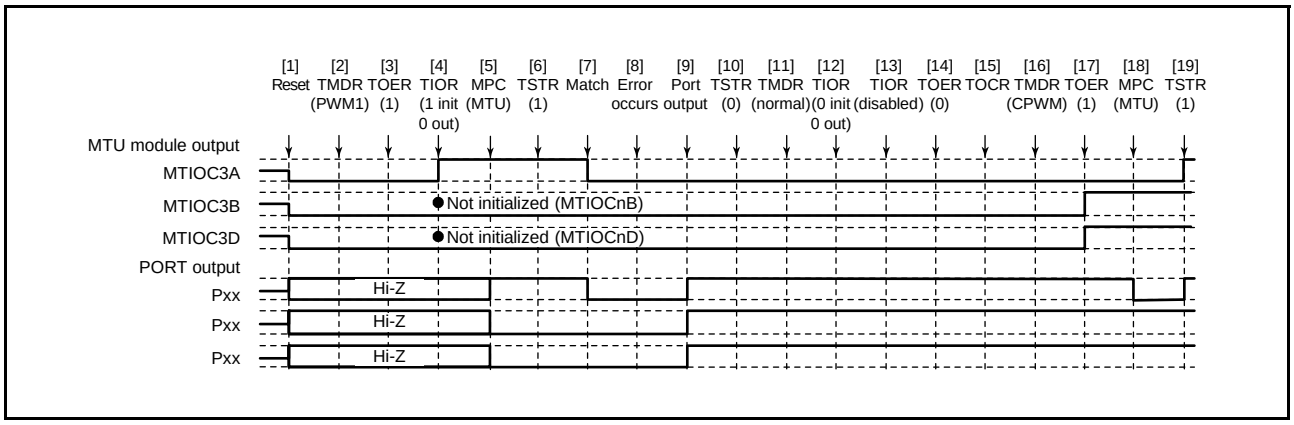



**Figure 20.135 Error Occurrence in PWM Mode 1, Recovery in Complementary PWM Mode**

[1] to [10] are the same as in Figure 20.131.
[11] Set normal mode to initialize the normal mode waveform generation section.
[12] Initialize the PWM mode 1 waveform generation section with TIOR.
[13] Disable operation of the PWM mode 1 waveform generation section with TIOR
[14] Disable output in MTU3 and MTU4 with TOER.
[15] Select the complementary PWM output level and enable or disable cyclic output with TOCR.
[16] Set complementary PWM mode.
[17] Enable output in MTU3 and MTU4 with TOER.
[18] Use the MPC and the port mode register (PMR) for the I/O port to set up MTU output.
[19] Restart operation by setting TSTR.

### (12) Operation When Error Occurs in PWM Mode 1 and Operation is Restarted in Reset-Synchronized PWM Mode

Figure 20.136 shows a case in which an error occurs in PWM mode 1 and operation is restarted in reset-synchronized PWM mode after re-setting.

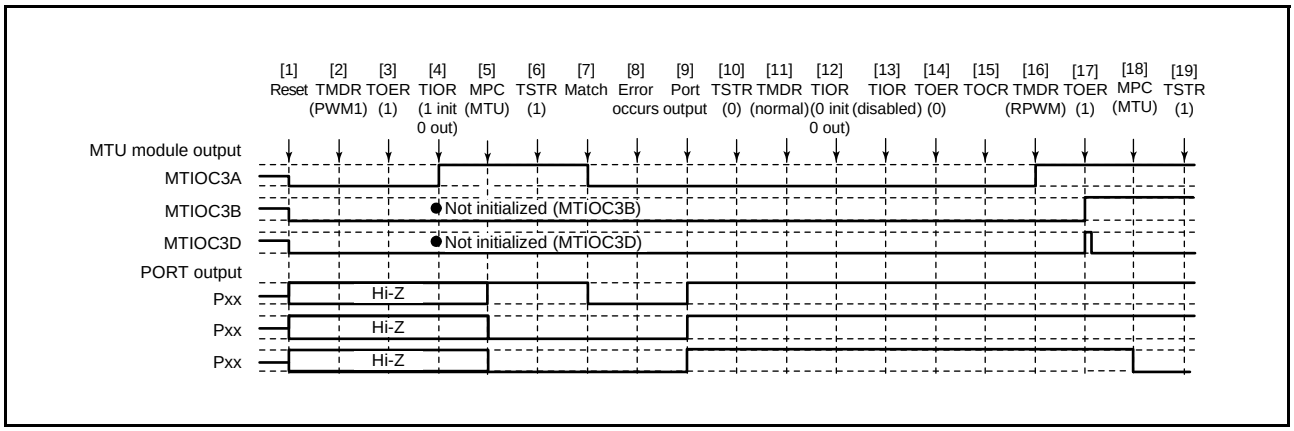



**Figure 20.136 Error Occurrence in PWM Mode 1, Recovery in Reset-Synchronized PWM Mode**

[1] to [14] are the same as in Figure 20.135.
[15] Select the reset-synchronized PWM output level and enable or disable cyclic output with TOCR.
[16] Set reset-synchronized PWM mode.
[17] Enable output in MTU3 and MTU4 with TOER.
[18] Use the MPC and the port mode register (PMR) for the I/O port to set up MTU output.
[19] Restart operation by setting TSTR.

### (13) Operation When Error Occurs in PWM Mode 2 and Operation is Restarted in Normal Mode

Figure 20.137 shows a case in which an error occurs in PWM mode 2 and operation is restarted in normal mode after re-setting.

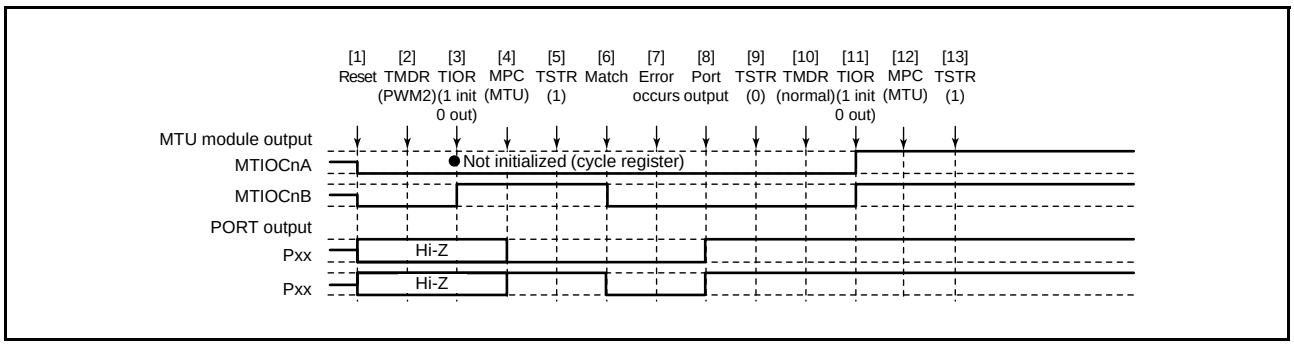



**Figure 20.137 Error Occurrence in PWM Mode 2, Recovery in Normal Mode**

[1] After a reset, the MTU output goes low and the ports enter high-impedance state.
[2] Set PWM mode 2.
[3] Initialize the pins with TIOR. (In the example, the initial output is a high level, and a low level is output on compare match occurrence. In PWM mode 2, the cycle register pins are not initialized. In the example, MTIOCnA is the cycle register.)
[4] Use the MPC and the port mode register (PMR) for the I/O port to set up MTU output.
[5] Start count operation by setting TSTR.
[6] Output goes low on compare match occurrence.
[7] An error occurs.
[8] Use the port direction register (PDR) and port mode register (PMR) for the input port pin to switch it to operate as a general output port pin, and the port output data register (PODR) to select output of the non-active level.
[9] Stop count operation by setting TSTR.
[10] Set normal mode.
[11] Initialize the pins with TIOR.
[12] Use the MPC and the port mode register (PMR) for the I/O port to set up MTU output.
[13] Restart operation by setting TSTR.

## (14) Operation When Error Occurs in PWM Mode 2 and Operation is Restarted in PWM Mode 1

Figure 20.138 shows a case in which an error occurs in PWM mode 2 and operation is restarted in PWM mode 1 after re-setting.

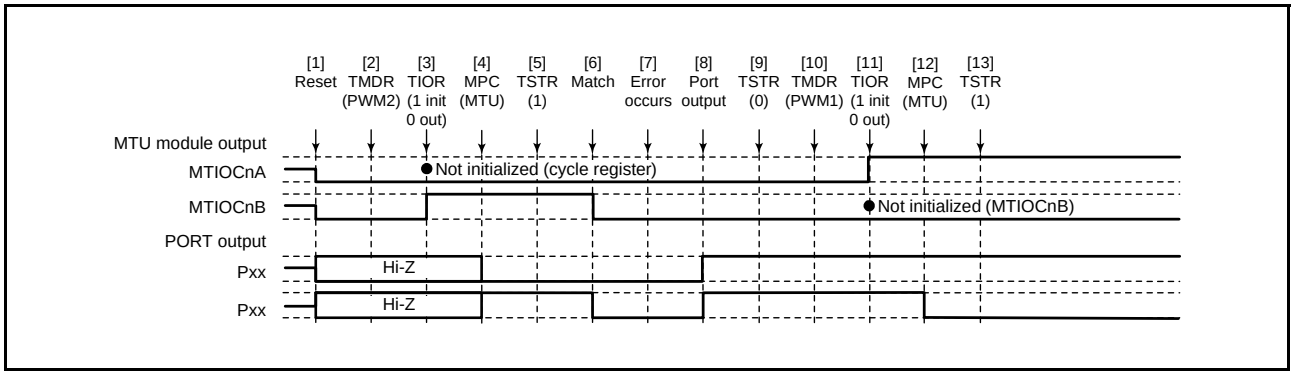



**Figure 20.138 Error Occurrence in PWM Mode 2, Recovery in PWM Mode 1**

[1] to [9] are the same as in Figure 20.137.
[10] Set PWM mode 1.
[11] Initialize the pins with the TIOR register. (In PWM mode 1, a waveform is not output on the MTIOCnB (MTIOCnD) pins. If a level is to be output, make the required general output port settings in the I/O port's port direction register (PDR) and port output data register (PODR).)
[12] Use the MPC and the port mode register (PMR) for the I/O port to set up MTU output.
[13] Restart operation by setting TSTR.

## (15) Operation When Error Occurs in PWM Mode 2 and Operation is Restarted in PWM Mode 2

Figure 20.139 shows a case in which an error occurs in PWM mode 2 and operation is restarted in PWM mode 2 after re-setting.

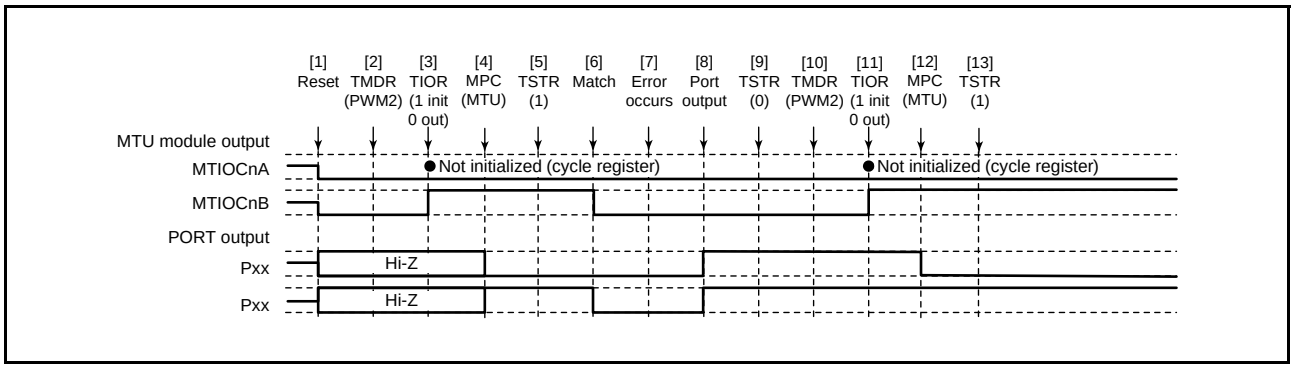



**Figure 20.139 Error Occurrence in PWM Mode 2, Recovery in PWM Mode 2**

[1] to [9] are the same as in Figure 20.137.
[10] This step is not necessary when restarting in PWM mode 2.
[11] Initialize the pins with the TIOR register. (In PWM mode 2, a waveform is not output on the cycle register pins. If a level is to be output, make the required general output port settings in the I/O port's port direction register (PDR) and port output data register (PODR).)
[12] Use the MPC and the port mode register (PMR) for the I/O port to set up MTU output.
[13] Restart operation by setting TSTR.

### (16) Operation When Error Occurs in PWM Mode 2 and Operation is Restarted in Phase Counting Mode

Figure 20.140 shows a case in which an error occurs in PWM mode 2 and operation is restarted in phase counting mode after re-setting.

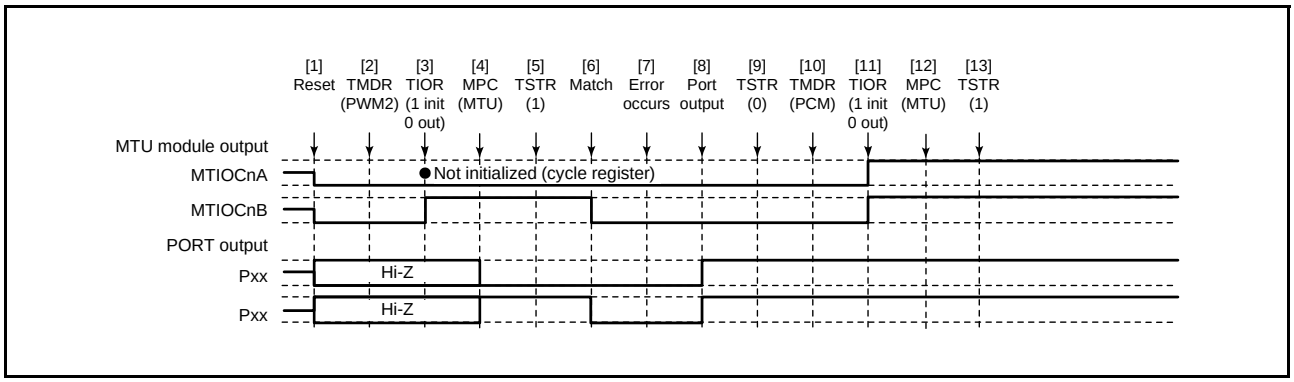



**Figure 20.140 Error Occurrence in PWM Mode 2, Recovery in Phase Counting Mode**

[1] to [9] are the same as in Figure 20.137.
[10] Set the phase counting mode.
[11] Initialize the pins with TIOR.
[12] Use the MPC and the port mode register (PMR) for the I/O port to set up MTU output.
[13] Restart operation by setting TSTR.

## (17) Operation When Error Occurs in Phase Counting Mode and Operation is Restarted in Normal Mode

Figure 20.141 shows a case in which an error occurs in phase counting mode and operation is restarted in normal mode after re-setting.

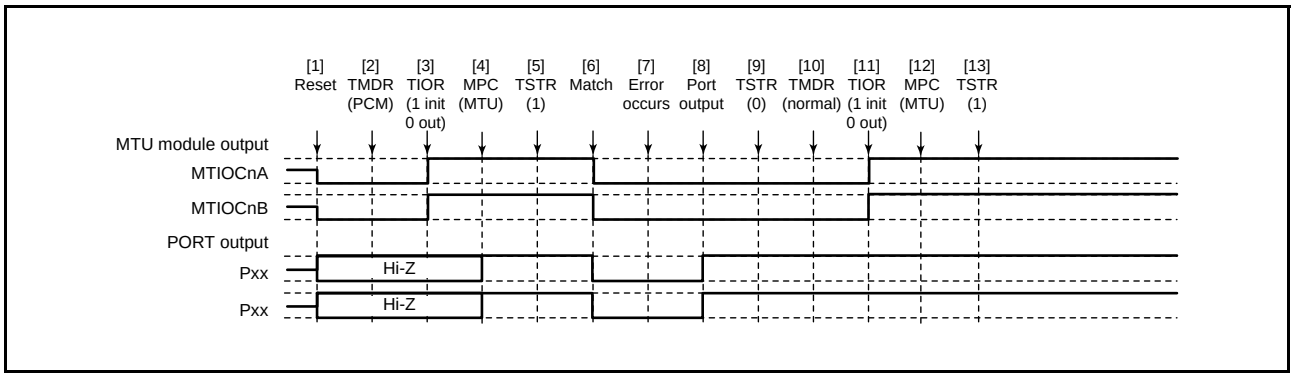



**Figure 20.141 Error Occurrence in Phase Counting Mode, Recovery in Normal Mode**

[1] After a reset, the MTU output goes low and the ports enter high-impedance state.
[2] Set phase counting mode.
[3] Initialize the pins with TIOR. (In the example, the initial output is a high level, and a low level is output on compare match occurrence.)
[4] Use the MPC and the port mode register (PMR) for the I/O port to set up MTU output.
[5] Start count operation by setting TSTR.
[6] Output goes low on compare match occurrence.
[7] An error occurs.
[8] Use the port direction register (PDR) and port mode register (PMR) for the input port pin to switch it to operate as a general output port pin, and the port output data register (PODR) to select output of the non-active level.
[9] Stop count operation by setting TSTR.
[10] Set normal mode.
[11] Initialize the pins with TIOR.
[12] Use the MPC and the port mode register (PMR) for the I/O port to set up MTU output.
[13] Restart operation by setting TSTR.

## (18) Operation When Error Occurs in Phase Counting Mode and Operation is Restarted in PWM Mode 1

Figure 20.142 shows a case in which an error occurs in phase counting mode and operation is restarted in PWM mode 1 after re-setting.

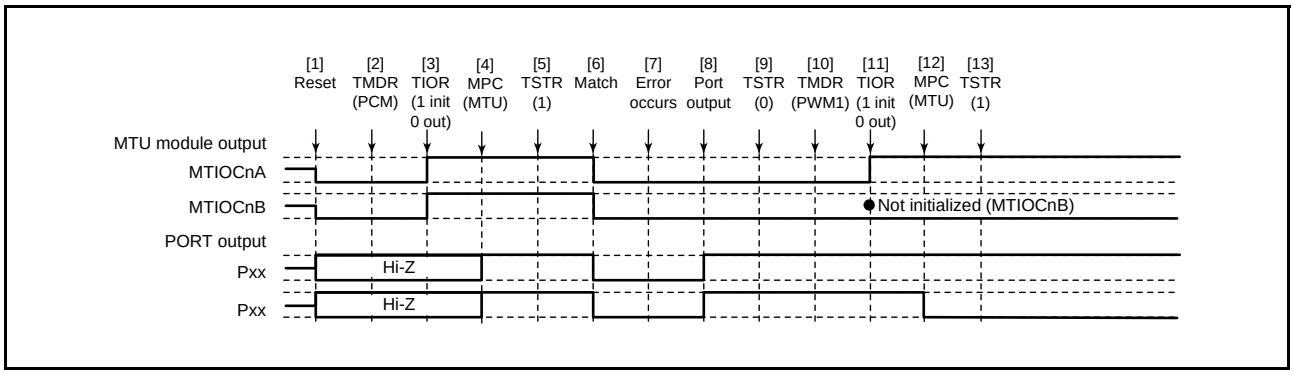



**Figure 20.142 Error Occurrence in Phase Counting Mode, Recovery in PWM Mode 1**

[1] to [9] are the same as in Figure 20.141.

[10] Set PWM mode 1.

[11] Initialize the pins with the TIOR register. (In PWM mode 1, a waveform is not output on the MTIOCnB (MTIOCnD) pins. If a level is to be output, make the required general output port settings in the I/O port's port direction register (PDR) and port output data register (PODR).)

[12] Use the MPC and the port mode register (PMR) for the I/O port to set up MTU output.

[13] Restart operation by setting TSTR.

### (19) Operation When Error Occurs in Phase Counting Mode and Operation is Restarted in PWM Mode 2

Figure 20.143 shows a case in which an error occurs in phase counting mode and operation is restarted in PWM mode 2 after re-setting.

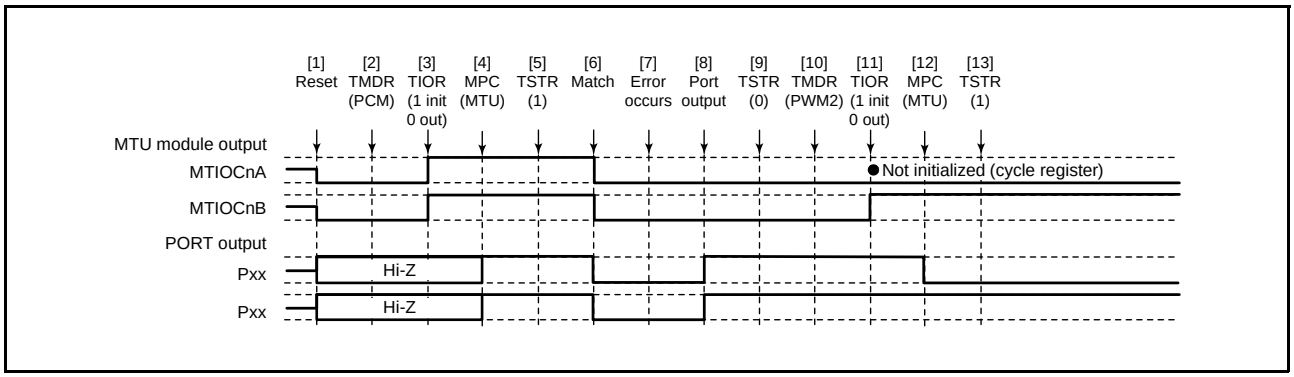



**Figure 20.143 Error Occurrence in Phase Counting Mode, Recovery in PWM Mode 2**

[1] to [9] are the same as in Figure 20.141.

[10] Set PWM mode 2.

[11] Initialize the pins with the TIOR register. (In PWM mode 1, a waveform is not output on the MTIOCnB (MTIOCnD) pins. If a level is to be output, make the required general output port settings in the I/O port's port direction register (PDR) and port output data register (PODR).)

[12] Use the MPC and the port mode register (PMR) for the I/O port to set up MTU output.

[13] Restart operation by setting TSTR.

## (20) Operation When Error Occurs in Phase Counting Mode and Operation is Restarted in Phase Counting Mode

Figure 20.144 shows a case in which an error occurs in phase counting mode and operation is restarted in phase counting mode after re-setting.

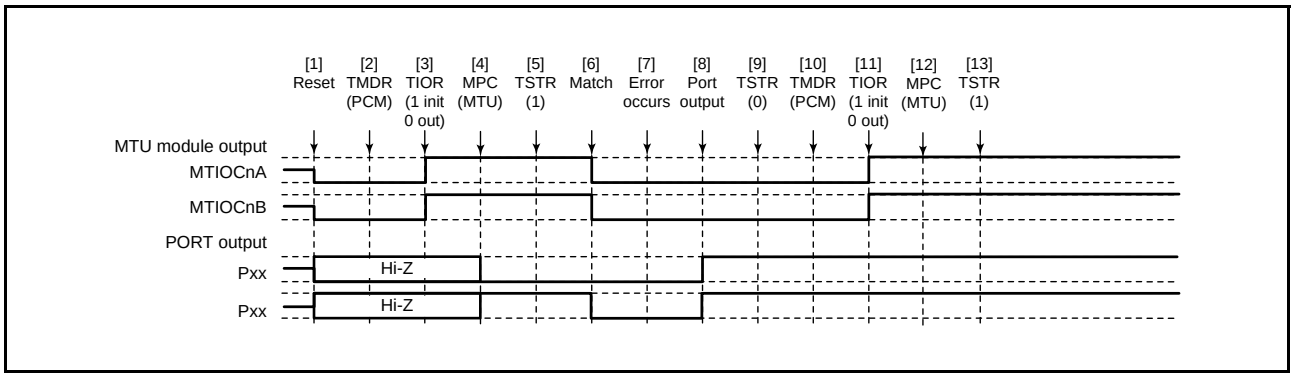



**Figure 20.144 Error Occurrence in Phase Counting Mode, Recovery in Phase Counting Mode**

[1] to [9] are the same as in Figure 20.141.
[10] This step is not necessary when restarting in phase counting mode.
[11] Initialize the pins with TIOR.
[12] Use the MPC and the port mode register (PMR) for the I/O port to set up MTU output.
[13] Restart operation by setting TSTR.

## (21) Operation When Error Occurs in Complementary PWM Mode and Operation is Restarted in Normal Mode

Figure 20.145 shows a case in which an error occurs in complementary PWM mode and operation is restarted in normal mode after re-setting.

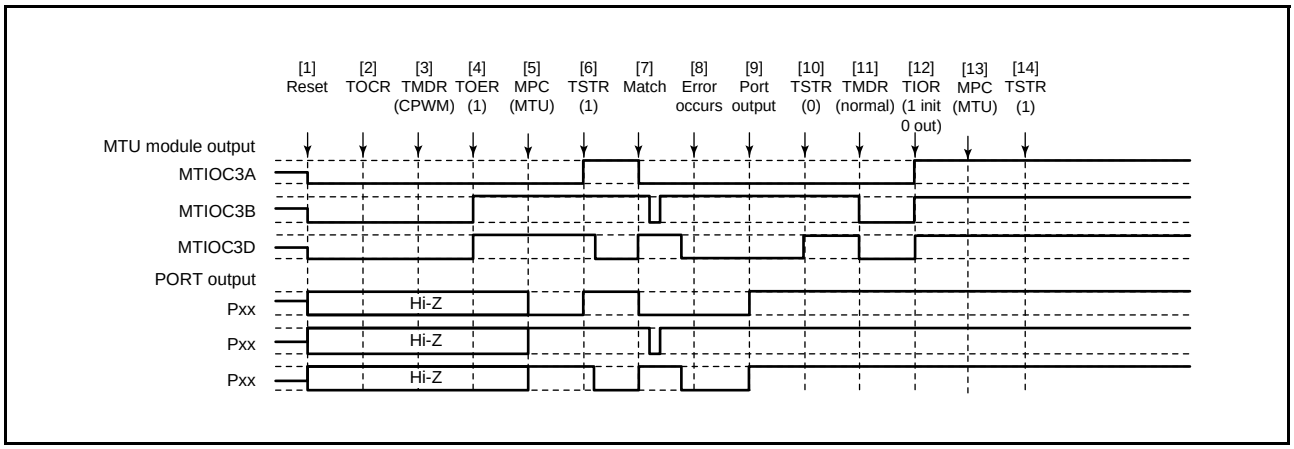



**Figure 20.145 Error Occurrence in Complementary PWM Mode, Recovery in Normal Mode**

[1] After a reset, the MTU output goes low and the ports enter high-impedance state.
[2] Select the complementary PWM output level and enable or disable cyclic output with TOCR.
[3] Set complementary PWM mode.
[4] Enable output in MTU3 and MTU4 with TOER.
[5] Use the MPC and the port mode register (PMR) for the I/O port to set up MTU output.
[6] Start count operation by setting TSTR.
[7] The complementary PWM waveform is output on compare match occurrence.
[8] An error occurs.
[9] Use the port direction register (PDR) and port mode register (PMR) for the input port pin to switch it to operate as a general output port pin, and the port output data register (PODR) to select output of the non-active level.
[10] Stop count operation by setting TSTR. (MTU output becomes the initial complementary PWM output value).
[11] Set normal mode (MTU output goes low).
[12] Initialize the pins with TIOR.
[13] Use the MPC and the port mode register (PMR) for the I/O port to set up MTU output.
[14] Restart operation by setting TSTR.

## (22) Operation When Error Occurs in Complementary PWM Mode and Operation is Restarted in PWM Mode 1

Figure 20.146 shows a case in which an error occurs in complementary PWM mode and operation is restarted in PWM mode 1 after re-setting.

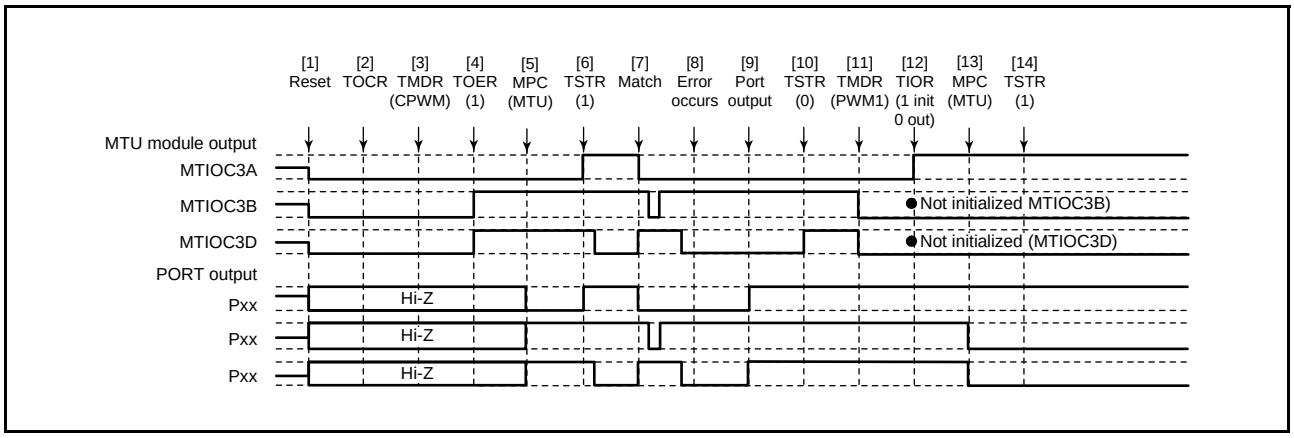



**Figure 20.146 Error Occurrence in Complementary PWM Mode, Recovery in PWM Mode 1**

[1] to [10] are the same as in Figure 20.145.

[11] Set PWM mode 1 (MTU output goes low).

[12] Initialize the pins with the TIOR register. (In PWM mode 1, a waveform is not output on the MTIOCnB (MTIOCnD) pins. If a level is to be output, make the required general output port settings in the I/O port's port direction register (PDR) and port output data register (PODR).)

[13] Use the MPC and the port mode register (PMR) for the I/O port to set up MTU output.

[14] Restart operation by setting TSTR.

### (23) Operation When Error Occurs in Complementary PWM Mode and Operation is Restarted in Complementary PWM Mode

Figure 20.147 shows a case in which an error occurs in complementary PWM mode and operation is restarted in complementary PWM mode after re-setting (when operation is restarted using the cycle and duty settings at the time of stopping the counter).

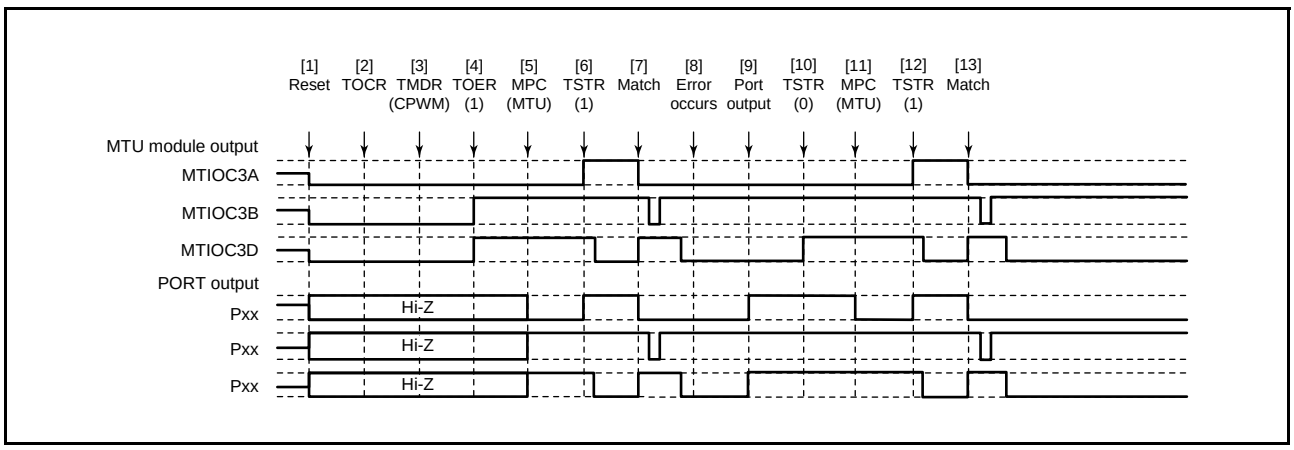



**Figure 20.147 Error Occurrence in Complementary PWM Mode, Recovery in Complementary PWM Mode**

[1] to [10] are the same as in Figure 20.145.
[11] Use the MPC and the port mode register (PMR) for the I/O port to set up MTU output.
[12] Restart operation by setting TSTR.
[13] The complementary PWM waveform is output on compare match occurrence.

### (24) Operation When Error Occurs in Complementary PWM Mode and Operation is Restarted in Complementary PWM Mode with New Settings

Figure 20.148 shows a case in which an error occurs in complementary PWM mode and operation is restarted in complementary PWM mode after re-setting (operation is restarted using new cycle and duty settings).

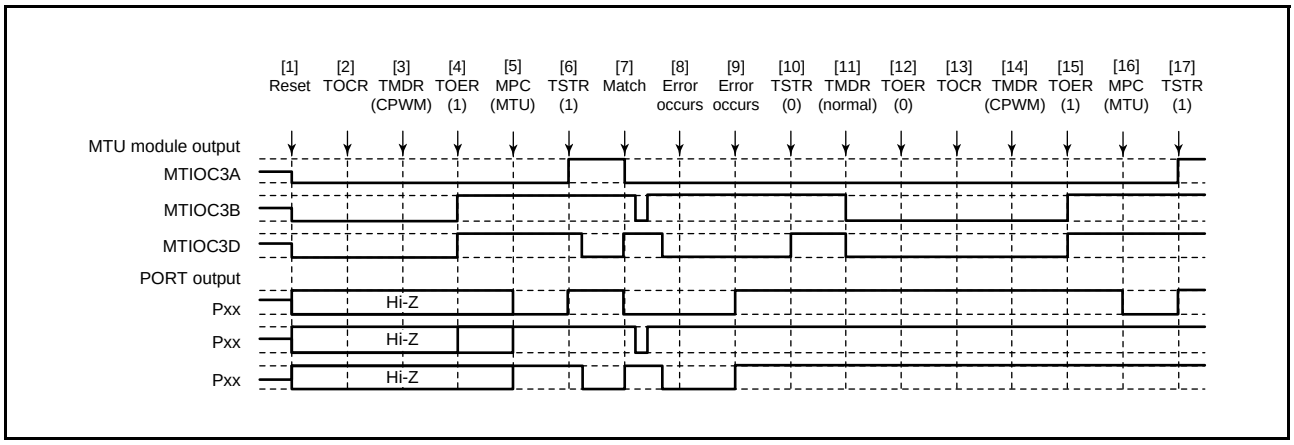



**Figure 20.148 Error Occurrence in Complementary PWM Mode, Recovery in Complementary PWM Mode**

[1] to [10] are the same as in Figure 20.145.
[11] Set normal mode and make new settings (MTU output goes low).
[12] Disable output in MTU3 and MTU4 with TOER.
[13] Select the complementary PWM output level and enable or disable cyclic output with TOCR.
[14] Set complementary PWM mode.
[15] Enable output in MTU3 and MTU4 with TOER.
[16] Use the MPC and the port mode register (PMR) for the I/O port to set up MTU output.
[17] Restart operation by setting TSTR.

### (25) Operation When Error Occurs in Complementary PWM Mode and Operation is Restarted in Reset-Synchronized PWM Mode

Figure 20.149 shows a case in which an error occurs in complementary PWM mode and operation is restarted in reset-synchronized PWM mode after re-setting.

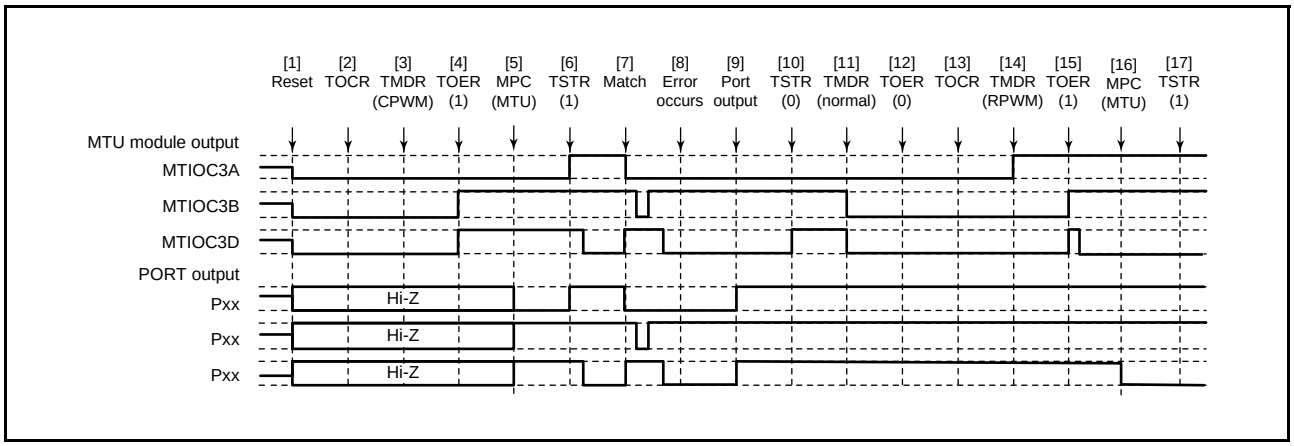



**Figure 20.149 Error Occurrence in Complementary PWM Mode, Recovery in Reset-Synchronized PWM Mode**

[1] to [10] are the same as in Figure 20.145.
[11] Set normal mode (MTU output goes low).
[12] Disable output in MTU3 and MTU4 with TOER.
[13] Select the reset-synchronized PWM output level and enable or disable cyclic output with TOCR.
[14] Set reset-synchronized PWM mode.
[15] Enable output in MTU3 and MTU4 with TOER.
[16] Use the MPC and the port mode register (PMR) for the I/O port to set up MTU output.
[17] Restart operation by setting TSTR.

## (26) Operation When Error Occurs in Reset-Synchronized PWM Mode and Operation is Restarted in Normal Mode

Figure 20.150 shows a case in which an error occurs in reset-synchronized PWM mode and operation is restarted in normal mode after re-setting.

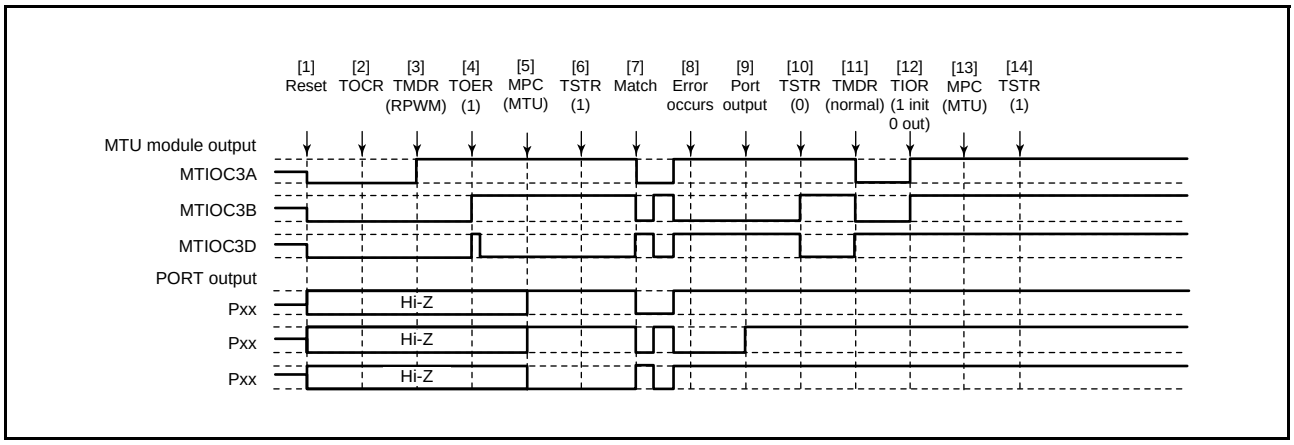



**Figure 20.150 Error Occurrence in Reset-Synchronized PWM Mode, Recovery in Normal Mode**

[1] After a reset, the MTU output goes low and the ports enter high-impedance state.
[2] Select the reset-synchronized PWM output level and enable or disable cyclic output with TOCR.
[3] Set reset-synchronized PWM mode.
[4] Enable output in MTU3 and MTU4 with TOER.
[5] Use the MPC and the port mode register (PMR) for the I/O port to set up MTU output.
[6] Start count operation by setting TSTR.
[7] The reset-synchronized PWM waveform is output on compare match occurrence.
[8] An error occurs.
[9] Use the port direction register (PDR) and port mode register (PMR) for the input port pin to switch it to operate as a general output port pin, and the port output data register (PODR) to select output of the non-active level.
[10] Stop count operation by setting TSTR. (MTU output becomes the initial reset-synchronized PWM output value.)
[11] Set normal mode (positive-phase MTU output goes low, and negative-phase output goes high).
[12] Initialize the pins with TIOR.
[13] Use the MPC and the port mode register (PMR) for the I/O port to set up MTU output.
[14] Restart operation by setting TSTR.

### (27) Operation When Error Occurs in Reset-Synchronized PWM Mode and Operation is Restarted in PWM Mode 1

Figure 20.151 shows a case in which an error occurs in reset-synchronized PWM mode and operation is restarted in PWM mode 1 after re-setting.

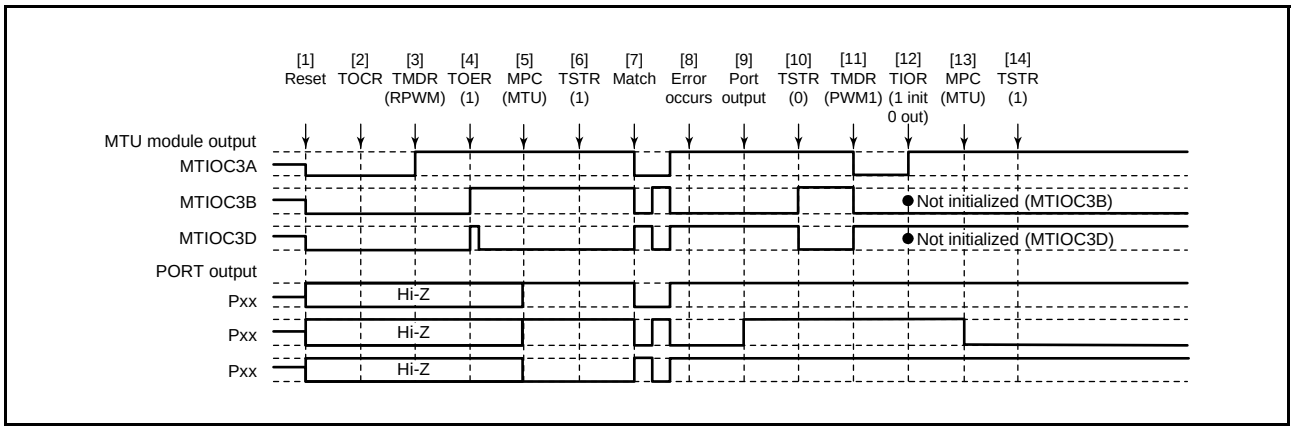



**Figure 20.151 Error Occurrence in Reset-Synchronized PWM Mode, Recovery in PWM Mode 1**

[1] to [10] are the same as in Figure 20.150.

[11] Set PWM mode 1 (positive-phase MTU output goes low, and negative-phase output goes high).

[12] Initialize the pins with the TIOR register. (In PWM mode 1, a waveform is not output on the MTIOCnB (MTIOCnD) pins. If a level is to be output, make the required general output port settings in the I/O port's port direction register (PDR) and port output data register (PODR).)

[13] Use the MPC and the port mode register (PMR) for the I/O port to set up MTU output.

[14] Restart operation by setting TSTR.

### (28) Operation When Error Occurs in Reset-Synchronized PWM Mode and Operation is Restarted in Complementary PWM Mode

Figure 20.152 shows a case in which an error occurs in reset-synchronized PWM mode and operation is restarted in complementary PWM mode after re-setting.

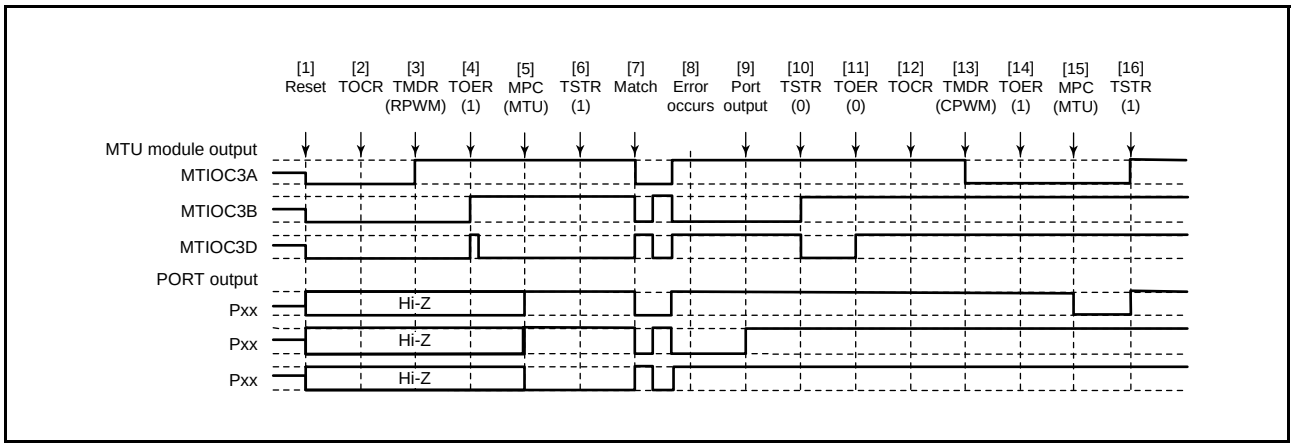



**Figure 20.152 Error Occurrence in Reset-Synchronized PWM Mode, Recovery in Complementary PWM Mode**

[1] to [10] are the same as in Figure 20.150.
[11] Disable output in MTU3 and MTU4 with TOER.
[12] Select the complementary PWM output level and enable or disable cyclic output with TOCR.
[13] Set complementary PWM mode (MTU cyclic output pin goes low).
[14] Enable output in MTU3 and MTU4 with TOER.
[15] Use the MPC and the port mode register (PMR) for the I/O port to set up MTU output.
[16] Restart operation by setting TSTR.

### (29) Operation When Error Occurs in Reset-Synchronized PWM Mode and Operation is Restarted in Reset-Synchronized PWM Mode

Figure 20.153 shows a case in which an error occurs in reset-synchronized PWM mode and operation is restarted in reset-synchronized PWM mode after re-setting.

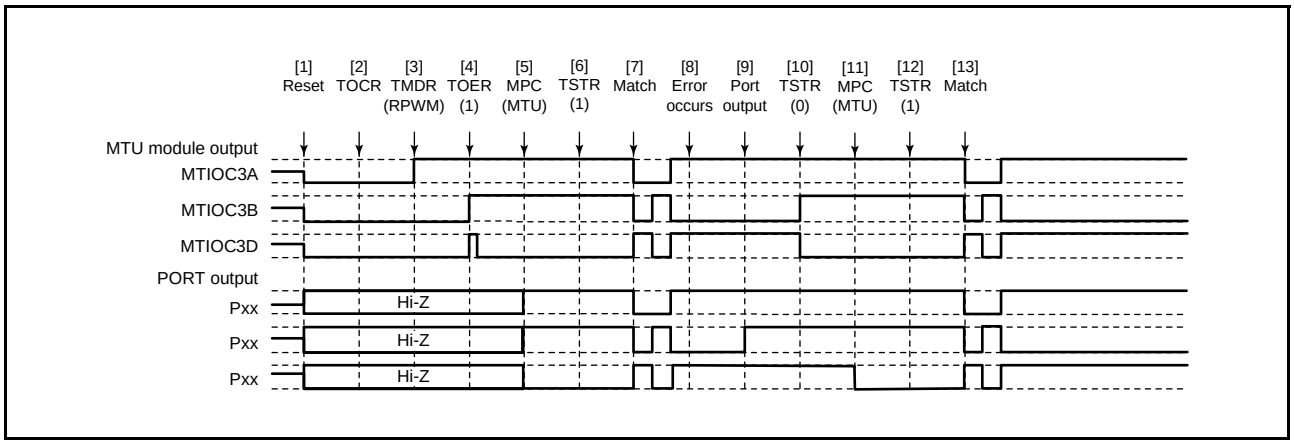



**Figure 20.153 Error Occurrence in Reset-Synchronized PWM Mode, Recovery in Reset-Synchronized PWM Mode**

[1] to [10] are the same as in Figure 20.150.

[11] Make MPC settings and port mode register (PMR) settings for the I/O port pins to operate as MTU outputs.

[12] Use the TSTR for a restart.

[13] The reset-synchronized PWM waveform is output on compare match occurrence.

## 20.8 Link Operation by ELC

### 20.8.1 Event Signal Output to ELC

MTU is capable of link operation for the preliminarily-setting module when interrupt request signal is used as an event signal by the event link controller (ELC).
The event signal is able to output regardless of the settings of the appropriate interrupt request enable bits.

### 20.8.2 MTU Operation by Event Signal Reception from ELC

MTU can be operated in the following by the preliminarily-setting event.

(1) Count start operation
The MTU is selected the count start operation when using the ELOPA and ELOPB registers setting of the ELC. The ELOPA register functions to channels 1 to 3, and ELOPB register functions to channel 4. TMDR of the channel set by MTU should be set to the value after reset, 00h. When the specified event is generated by the ELSRn register, the TSTR.CSTn bit shown in Table 20.60 is set to 1, then the MTU counter is started.
However, when the specified event is generated while TSTR.CSTn bit is set to 1, the event is disabled. Table 20.60 lists the TSTR register bits used for each channel.
For details on the count start operation setting, refer to section 20.3.1, (1) Counter Operation.

**Table 20.60 Linkage Operating TSTR Register by ELC**

| Channel No. | TSTR register |
|---|---|
| Channel 1 | TSTR.CST1 bit |
| Channel 2 | TSTR.CST2 bit |
| Channel 3 | TSTR.CST3 bit |
| Channel 4 | TSTR.CST4 bit |

(2) Input capture operation
The MTU is selected the input capture operation when using the ELOPA and ELOPB registers setting of the ELC. The ELOPA register handles channels 1 to 3, and ELOPB register handles channel 4. TMDR of the channel set by MTU should be set to the value after reset, 00h. When the specified event is generated by the ELSRn register, then the TCNT counter value capture to TGR register. When using the input capture operation, after setting the bit of MTU TIOR register to the input capture, TSTR.CSTn bit should be set to 1, and start the counter.
Then, the TIOCnA pin (input capture pin) input is disabled.
Table 20.61 lists the timer general register and timer I/O control register used in the input capture operation by ELC.
For details on the input capture setting, refer to section 20.3.1, (3) Input Capture Function.

**Table 20.61 Timer General Register and Timer I/O Control Register used in the Input Capture Operation by ELC**

| Channel No. | Register Name | Bit Name of TIOR Register |
|---|---|---|
| Channel 1 | TGRA register | TIOR.IOA[3:0] bits |
| Channel 2 | TGRA register | TIOR.IOA[3:0] bits |
| Channel 3 | TGRA register | TIORH.IOA[3:0] bits |
| Channel 4 | TGRA register | TIORH.IOA[3:0] bits |

(3) Counter restart operation

The MTU is selected the count start operation when using the ELOPA and ELOPB registers setting of the ELC. The ELOPA register functions channels 1 to 3, and ELOPB register functions channel 4. TMDR of the channel set by MTU should be set to the value after reset, 00h. When the specified event is generated by the ELSRn register, then the TCNT (timer counter register) value is rewritten to initial value. In the CSTn bit of the TSTR register is set to 1, can be continued to the count operation. For details on the TSTR.CSTn bit, see Table 20.60.

### 20.8.3 Usage Notes on MTU by Event Signal Reception from ELC

The following describes usage notes when using MTU by the event link operation.

(1) Count start operation

When the specified event is generated by the ELSRn register while write cycle is performed to the TSTR.CSTn bit, the write cycle is not performed to the TSTR.CSTn bit, and the setting to 1 takes precedence by generated event.

(2) Count restart operation

When the specified event is generated by the ELSRn register while write cycle is performed to the TCNT counter, the write cycle is not performed to the TCNT counter, and count value initialization takes precedence by generated event.

# 21. Port Output Enable 2 (POE2a)

The port output enable 2 (POE) module can be used to place the states of the pins for complementary PWM output by the MTU (MTIOC3B, MTIOC3D, MTIOC4A, MTIOC4B, MTIOC4C, and MTIOC4D), and the states of pins for MTU0 (MTIOC0A, MTIOC0B, MTIOC0C, and MTIOC0D) in the high-impedance in response to changes in the input levels on the POE0# to POE3# and POE8# pins, in the output levels on pins for complementary PWM output by the MTU, oscillation stop detection by the clock generation circuit, and changes to register settings (SPOER).
It can also generate simultaneous interrupt requests.

## 21.1 Overview

Table 21.1 lists the specifications of the POE, and Figure 21.1 shows a block diagram of the POE.

**Table 21.1 POE Specifications**

| Item | Description |
|---|---|
| High-impedance is controlled by the input level detection | • Falling-edge detection or sampling of the low level 16 times at PCLK/8, PCLK/16, or PCLK/128 clock cycles can be set for each of the POE0# to POE3# and POE8# input pins.<br>• Pins for complementary PWM output from the MTU can be placed in the high-impedance on detection of falling edges or sampling of the low level on the POE0# to POE3# pins.<br>• Pins for output from MTU0 can be placed in the high-impedance on detection of falling edges or sampling of the low level on the POE8# pin. |
| High-impedance is controlled by the output level comparison | • Levels output on pins for complementary PWM output from the MTU are compared, and when simultaneous output of the active level continues for one or more clock cycles, the pins can be placed in the high-impedance. |
| High-impedance is controlled by the oscillation stop detection | • Pins for complementary PWM output from the MTU and output pins for MTU0 can be placed in the high-impedance when oscillation by the clock generation circuit stops. |
| High-impedance is controlled by software (registers) | • Pins for complementary PWM output from the MTU and output pins for MTU0 can be placed in the high-impedance by modifying settings of POE registers. |
| Interrupts | • Interrupts can be generated in response to the results of POE0# to POE3# and POE8# input-level detection and MTU complementary PWM output-level comparison. |

The POE has input-level detection circuits, output-level comparison circuits, an input for the oscillation stop detection signal from the clock generation circuit, and a high-impedance request/interrupt request generating circuit as shown in Figure 21.1.

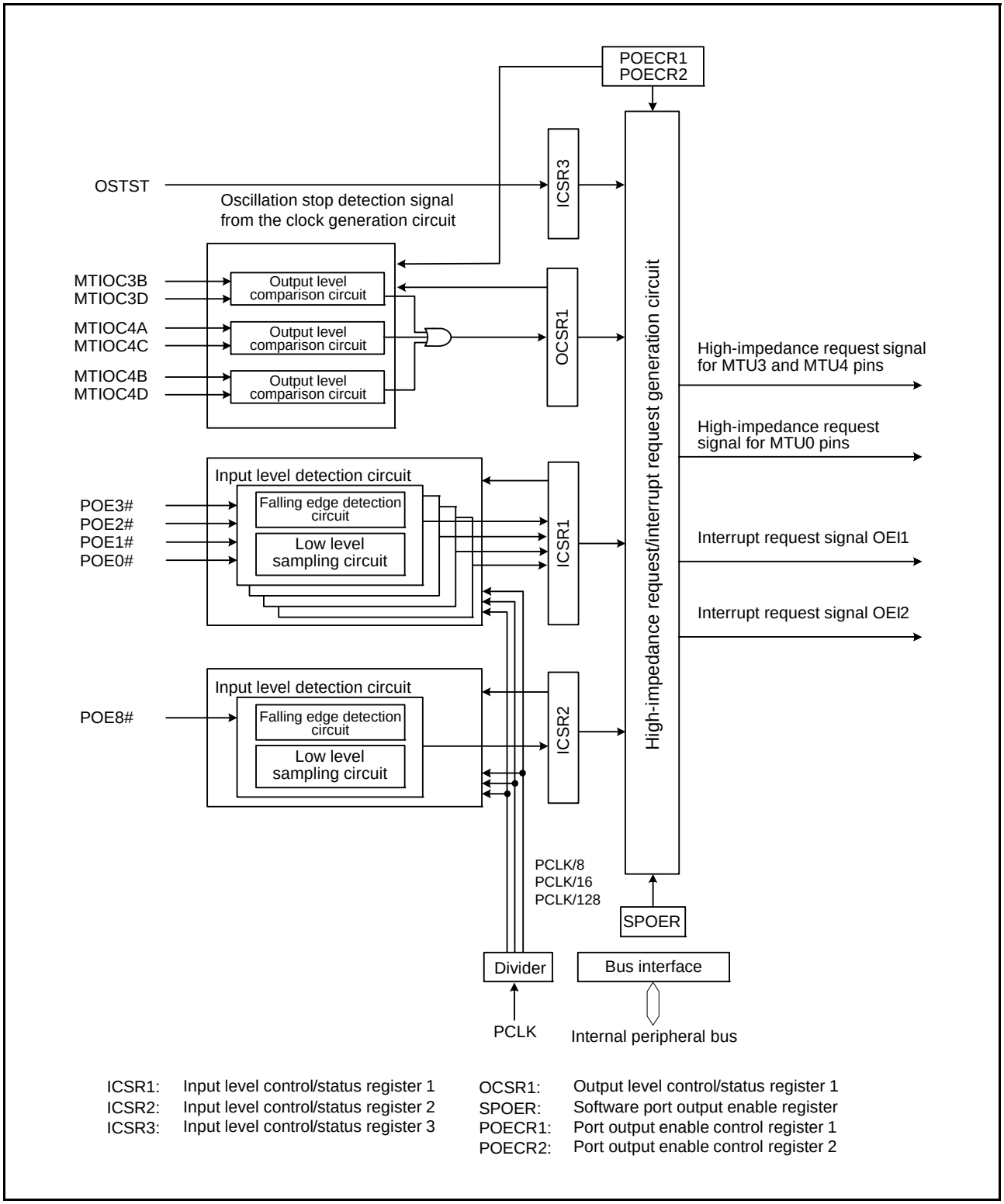



**Figure 21.1 POE Block Diagram**

Table 21.2 lists I/O pins to be used by the POE.

**Table 21.2 POE I/O Pins**

| Pin Name | I/O | Description |
|---|---|---|
| POE0# to POE3# | Input | Request signals to place the pins for MTU complementary PWM output in high-impedance. |
| POE8# | Input | Request signal to place the MTU0 output pins in high-impedance. |
| MTIOC3B | Output | MTU3 complementary PWM output pin |
| MTIOC3D | Output | MTU3 complementary PWM output pin |
| MTIOC4A | Output | MTU4 complementary PWM output pin |
| MTIOC4B | Output | MTU4 complementary PWM output pin |
| MTIOC4C | Output | MTU4 complementary PWM output pin |
| MTIOC4D | Output | MTU4 complementary PWM output pin |
| MTIOC0A | Output | MTU0 output pin |
| MTIOC0B | Output | MTU0 output pin |
| MTIOC0C | Output | MTU0 output pin |
| MTIOC0D | Output | MTU0 output pin |

Table 21.3 lists output-level comparisons with pin combinations.

**Table 21.3 Pin Combinations**

| Pin Combination | I/O | Description |
|---|---|---|
| MTIOC3B and MTIOC3D | Output | Pin combinations for output-level comparison and high-impedance control can be selected by POE registers.<br>The pins for MTU complementary PWM output are placed in high-impedance when the pins simultaneously output an active level for one or more PCLK clock cycles.<br>(When the MTU. TOCR1.TOCS bit = 0:<br>The active level is low level if the MTU.TOCR1.OLSP and OLSN bits are 0, and the active level is high level if the MTU.TOCR1.OLSP and OLSN bits are 1.<br>When the MTU. TOCR1.TOCS bit = 1:<br>The active level is low level if the MTU.TOCR2.OLS3N, OLS3P, OLS2N, OLS2P, OLS1N and OLS1P bits are 0, and the active level is high level if the MTU.TOCR2.OLS3N, OLS3P, OLS2N, OLS2P, OLS1N and OLS1P bits are 1.) |
| MTIOC4A and MTIOC4C | Output | |
| MTIOC4B and MTIOC4D | Output | |

## 21.2 Register Descriptions

### 21.2.1 Input Level Control/Status Register 1 (ICSR1)

Address(es): 0008 8900h

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | POE3F | POE2F | POE1F | POE0F | — | — | — | PIE1 | POE3M[1:0] | | POE2M[1:0] | | POE1M[1:0] | | POE0M[1:0] | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b1, b0 | POE0M[1:0] | POE0 Mode Select | b1 b0<br>0 0: Accepts a high-impedance request on the falling edge of the POE0# pin input.<br>0 1: Accepts a high-impedance request when the POE0# pin input has been sampled 16 times at PCLK/8 clock cycles and all are low level.<br>1 0: Accepts a request when POE0# input has been sampled 16 times at PCLK/16 clock pulses and all are low level.<br>1 1: Accepts a request when POE0# input has been sampled 16 times at PCLK/128 clock pulses and all are low level. | R/W*1 |
| b3, b2 | POE1M[1:0] | POE1 Mode Select | b3 b2<br>0 0: Accepts a high-impedance request on the falling edge of the POE1# pin input.<br>0 1: Accepts a high-impedance request when the POE1# pin input has been sampled 16 times at PCLK/8 clock cycles and all are low level.<br>1 0: Accepts a request when POE1# input has been sampled 16 times at PCLK/16 clock pulses and all are low level.<br>1 1: Accepts a request when POE1# input has been sampled 16 times at PCLK/128 clock pulses and all are low level. | R/W*1 |
| b5, b4 | POE2M[1:0] | POE2 Mode Select | b5 b4<br>0 0: Accepts a high-impedance request on the falling edge of the POE2# pin input.<br>0 1: Accepts a high-impedance request when the POE2# pin input has been sampled 16 times at PCLK/8 clock cycles and all are low level.<br>1 0: Accepts a request when POE2# input has been sampled 16 times at PCLK/16 clock pulses and all are low level.<br>1 1: Accepts a request when POE2# input has been sampled 16 times at PCLK/128 clock pulses and all are low level. | R/W*1 |
| b7, b6 | POE3M[1:0] | POE3 Mode Select | b7 b6<br>0 0: Accepts a high-impedance request on the falling edge of the POE3# pin input.<br>0 1: Accepts a high-impedance request when the POE3# pin input has been sampled 16 times at PCLK/8 clock cycles and all are low level.<br>1 0: Accepts a request when POE3# input has been sampled 16 times at PCLK/16 clock pulses and all are low level.<br>1 1: Accepts a request when POE3# input has been sampled 16 times at PCLK/128 clock pulses and all are low level. | R/W*1 |
| b8 | PIE1 | Port Interrupt Enable 1 | 0: OEI1 interrupt requests by the input level detection disabled<br>1: OEI1 interrupt requests by the input level detection enabled | R/W |
| b11 to b9 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |
| b12 | POE0F | POE0 Flag | 0: Indicates that a high-impedance request has not been input to the POE0# pin.<br>1: Indicates that a high-impedance request has been input to the POE0# pin. | R/(W)*2 |
| b13 | POE1F | POE1 Flag | 0: Indicates that a high-impedance request has not been input to the POE1# pin.<br>1: Indicates that a high-impedance request has been input to the POE1# pin. | R/(W)*2 |
| b14 | POE2F | POE2 Flag | 0: Indicates that a high-impedance request has not been input to the POE2# pin.<br>1: Indicates that a high-impedance request has been input to the POE2# pin. | R/(W)*2 |
| b15 | POE3F | POE3 Flag | 0: Indicates that a high-impedance request has not been input to the POE3# pin.<br>1: Indicates that a high-impedance request has been input to the POE3# pin. | R/(W)*2 |

Note 1. Can be modified only once after a reset.
Note 2. Writing 0 to this bit after reading it as 1 clears the flag and is the only allowed way.

When low-level sampling has been set by the POE0M[1:0] to POE3M[1:0] bits, writing 0 to the POE0F to POE3F flags requires high-level input on the POE0# to POE3# pins.
For details, refer to section 21.3.5, Release from the High-Impedance.

**PIE1 Bit (Port Interrupt Enable 1)**

This bit enables or disables OEI1 interrupt requests when any one of the POE0F to POE3F flags is set to 1.

**POE0F Flag (POE0 Flag)**

This flag indicates that a high-impedance request has been input to the POE0# pin.
[Clearing condition]

- By writing 0 to POE0F after reading POE0F = 1

[Setting condition]

- When the input set by POE0M[1:0] occurs at the POE0# pin

**POE1F Flag (POE1 Flag)**

This flag indicates that a high-impedance request has been input to the POE1# pin.
[Clearing condition]

- By writing 0 to POE1F after reading POE1F = 1

[Setting condition]

- When the input set by POE1M[1:0] occurs at the POE1# pin

**POE2F Flag (POE2 Flag)**

This flag indicates that a high-impedance request has been input to the POE2# pin.
[Clearing condition]

- By writing 0 to POE2F after reading POE2F = 1

[Setting condition]

- When the input set by POE2M[1:0] occurs at the POE2# pin

**POE3F Flag (POE3 Flag)**

This flag indicates that a high-impedance request has been input to the POE3# pin.
[Clearing condition]

- By writing 0 to POE3F after reading POE3F = 1

[Setting condition]

- When the input set by POE3M[1:0] occurs at the POE3# pin

## 21.2.2 Output Level Control/Status Register 1 (OCSR1)

Address(es): 0008 8902h

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | OSF1 | — | — | — | — | — | OCE1 | OIE1 | — | — | — | — | — | — | — | — |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b7 to b0 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |
| b8 | OIE1 | Output Short Interrupt Enable 1 | 0: OEI1 interrupt requests by the output level comparison disabled<br>1: OEI1 interrupt requests by the output level comparison enabled | R/W |
| b9 | OCE1 | Output Short High-Impedance Enable 1 | 0: Does not place the pins in high-impedance.<br>1: Places the pins in high-impedance. | R/W*1 |
| b14 to b10 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |
| b15 | OSF1 | Output Short Flag 1 | 0: Indicates that outputs have not simultaneously become an active level.<br>1: Indicates that outputs have simultaneously become an active level. | R/(W)*2 |

Note 1. Can be modified only once after a reset.
Note 2. Writing 0 to this bit after reading it as 1 clears the flag and is the only allowed way.

### OIE1 Bit (Output Short Interrupt Enable 1)

This bit enables or disables OEI1 interrupt requests when the OSF1 flag is set to 1.

### OCE1 Bit (Output Short High-Impedance Enable 1)

This bit specifies whether to place the MTU complementary PWM output pins in high-impedance when the OSF1 flag is set to 1.

### OSF1 Flag (Output Short Flag 1)

This flag indicates that any one of the three pairs of two-phase outputs for MTU complementary PWM output to be compared in Table 21.3 has simultaneously become an active level. If the POECR2.PnCZEA (n = 1, 2, 3) bits are 0 or the output comparison function of the MTU is not enabled, the OSF1 flag will not be set to 1 even if both pins in the corresponding complementary output pair of the MTU are simultaneously active.

[Clearing condition]

- By writing 0 to OSF1 after reading OSF1 = 1

The complementary output pins for the MTU must be at the inactive level when 0 is written to the flag.

For details, refer to section 21.3.5, Release from the High-Impedance.

[Setting condition]

- When any one of the three pairs of two-phase outputs has simultaneously become an active level

## 21.2.3 Input Level Control/Status Register 2 (ICSR2)

Address(es): 0008 8908h

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | POE8F | — | — | POE8E | PIE2 | — | — | — | — | — | — | POE8M[1:0] | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b1, b0 | POE8M[1:0] | POE8 Mode Select | b1 b0<br>0 0: Accepts a high-impedance request on the falling edge of the POE8# pin input<br>0 1: Accepts a high-impedance request when the POE8# pin input has been sampled 16 times at PCLK/8 clock cycles and all are low level.<br>1 0: Accepts a high-impedance request when the POE8# pin input has been sampled 16 times at PCLK/16 clock cycles and all are low level.<br>1 1: Accepts a high-impedance request when the POE8# pin input has been sampled 16 times at PCLK/128 clock cycles and all are low level. | R/W*1 |
| b7 to b2 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |
| b8 | PIE2 | Port Interrupt Enable 2 | 0: OEI2 interrupt requests disabled<br>1: OEI2 interrupt requests enabled | R/W |
| b9 | POE8E | POE8 High-Impedance Enable | 0: Does not place the MTIOC0A, MTIOC0B, MTIOC0C, and MTIOC0D pins in high-impedance.<br>1: Places the MTIOC0A, MTIOC0B, MTIOC0C, and MTIOC0D pins in high-impedance. | R/W*1 |
| b11, b10 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |
| b12 | POE8F | POE8 Flag | 0: Indicates that a high-impedance request has not been input to the POE8# pin.<br>1: Indicates that a high-impedance request has been input to the POE8# pin. | R/(W)*2 |
| b15 to b13 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |

Note 1. Can be modified only once after a reset.
Note 2. Writing 0 to this bit after reading it as 1 clears the flag and is the only allowed way.

### PIE2 Bit (Port Interrupt Enable 2)

This bit enables or disables OEI2 interrupt requests when the POE8F flag is set to 1.

### POE8E Bit (POE8 High-Impedance Enable)

This bit specifies whether to place the MTU0 pins in high-impedance when the POE8F flag is set to 1.

### POE8F Flag (POE8 Flag)

This flag indicates that a high-impedance request has been input to the POE8# pin.

[Clearing conditions]

Writing 0 to POE8F after reading POE8F = 1

When writing 0 to the flag while low-level sampling is selected for the POE8M[1:0] bits, the POE8# pin input must be at the high level.

For details, refer to section 21.3.5, Release from the High-Impedance.

[Setting condition]

- When the input set by POE8M[1:0] occurs at the POE8# pin

## 21.2.4 Software Port Output Enable Register (SPOER)

Address(es): 0008 890Ah

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | CH0HIZ | CH34HIZ |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b0 | CH34HIZ | MTU3 and MTU4 Output High-Impedance Enable | 0: Does not place the pins in high-impedance.<br>1: Places the pins in high-impedance. | R/W |
| b1 | CH0HIZ | MTU0 Output High-Impedance Enable | 0: Does not place the pins in high-impedance.<br>1: Places the pins in high-impedance. | R/W |
| b7 to b2 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |

**CH34HIZ Bit (MTU3 and MTU4 Output High-Impedance Enable)**

This bit selects whether to place the MTU complementary PWM output pins (MTIOC3B, MTIOC3D, MTIOC4A, MTIOC4B, MTIOC4C, MTIOC4D) in high-impedance.

[Clearing condition]

- By writing 0 to CH34HIZ after reading CH34HIZ = 1

[Setting condition]

- By writing 1 to CH34HIZ

**CH0HIZ Bit (MTU0 Output High-Impedance Enable)**

This bit selects whether to place the MTU0 pins (MTIOC0A, MTIOC0B, MTIOC0C, MTIOC0D) in high-impedance.

[Clearing condition]

- By writing 0 to CH0HIZ after reading CH0HIZ = 1

[Setting condition]

- By writing 1 to CH0HIZ

## 21.2.5 Port Output Enable Control Register 1 (POECR1)

Address(es): 0008 890Bh

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | PE3ZE | PE2ZE | PE1ZE | PE0ZE |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b0 | PE0ZE | MTIOC0A High-Impedance Enable | 0: Does not place the pin in high-impedance.<br>1: Places the pin in high-impedance. | R/W*1 |
| b1 | PE1ZE | MTIOC0B High-Impedance Enable | 0: Does not place the pin in high-impedance.<br>1: Places the pin in high-impedance. | R/W*1 |
| b2 | PE2ZE | MTIOC0C High-Impedance Enable | 0: Does not place the pin in high-impedance.<br>1: Places the pin in high-impedance. | R/W*1 |
| b3 | PE3ZE | MTIOC0D High-Impedance Enable | 0: Does not place the pin in high-impedance.<br>1: Places the pin in high-impedance. | R/W*1 |
| b7 to b4 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |

Note 1. Can be modified only once after a reset.

## 21.2.6 Port Output Enable Control Register 2 (POECR2)

Address(es): 0008 890Ch

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | P1CZEA | P2CZEA | P3CZEA | — | — | — | — |
| Value after reset: | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b3 to b0 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |
| b4 | P3CZEA | MTU Port 3 High-Impedance Enable | 0: Comparison of output levels does not proceed and the pins are not placed in the high-impedance.<br>1: The pins are placed in the high-impedance. | R/W*1 |
| b5 | P2CZEA | MTU Port 2 High-Impedance Enable | 0: Comparison of output levels does not proceed and the pins are not placed in the high-impedance.<br>1: The pins are placed in the high-impedance. | R/W*1 |
| b6 | P1CZEA | MTU Port 1 High-Impedance Enable | 0: Comparison of output levels does not proceed and the pins are not placed in the high-impedance.<br>1: The pins are placed in the high-impedance. | R/W*1 |
| b7 | — | Reserved | This bit is read as 0. The write value should be 0. | R/W |

Note 1. Can be modified only once after a reset.

When this function is not used, write 00h to this register.

### P3CZEA Bit (MTU Port 3 High-Impedance Enable)

This bit gives permission regarding whether or not the MTIOC4B and MTIOC4D pins for complementary PWM output from the MTU are placed in the high-impedance. It also gives permission regarding whether or not the levels on the MTIOC4B and MTIOC4D pins are compared.

### P2CZEA Bit (MTU Port 2 High-Impedance Enable)

This bit gives permission regarding whether or not the MTIOC4A and MTIOC4C pins for complementary PWM output from the MTU are placed in the high-impedance. It also gives permission regarding whether or not the levels on the MTIOC4A and MTIOC4C pins are compared.

### P1CZEA Bit (MTU Port 1 High-Impedance Enable)

This bit gives permission regarding whether or not the MTIOC3B and MTIOC3D pins for complementary PWM output from the MTU are placed in the high-impedance. It also gives permission regarding whether or not the levels on the MTIOC3B and MTIOC3D pins are compared.

## 21.2.7 Input Level Control/Status Register 3 (ICSR3)

Address(es): 0008 890Eh

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | OSTSTF | — | — | OSTSTE | — | — | — | — | — | — | — | — | — |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b8 to b0 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |
| b9 | OSTSTE | OSTST High-Impedance Enable | 0: Does not place the MTIOC0A, MTIOC0B, MTIOC0C, MTIOC0D, MTIOC3B, MTIOC3D, MTIOC4A, MTIOC4B, MTIOC4C, and MTIOC4D pins in high-impedance.<br>1: Places the MTIOC0A, MTIOC0B, MTIOC0C, MTIOC0D, MTIOC3B, MTIOC3D, MTIOC4A, MTIOC4B, MTIOC4C, and MTIOC4D pins in high-impedance. | R/W*1 |
| b11, b10 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |
| b12 | OSTSTF | OSTST High-Impedance Flag | 0: Oscillation stop is not producing a request to place pins in the high-impedance.<br>1: Oscillation stop is producing a request to place pins in the high-impedance. | R/(W)*2 |
| b15 to b13 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |

Note 1. Can be modified only once after a reset.
Note 2. Writing 0 to this bit after reading it as 1 clears the flag and is the only allowed way.

### OSTSTE Bit (OSTST High-Impedance Enable)

This bit permits or prohibits placement of pins for complementary PWM output from MTU and output pins for MTU0 in the high-impedance on detection that oscillation has stopped.

### OSTSTF Flag (OSTST High-Impedance Flag)

The OSTSTF flag is a status flag that indicates the state of requests to place pins in the high-impedance due to oscillation having stopped. The value of the flag becomes 1 when oscillation stops. Ensure that the oscillation-stopped detection signal is negated when clearing the flag by writing 0 to it. Writing 0 to the OSTSTF flag will not clear the flag while the oscillation-stopped detection signal is being asserted; in other words, it will not clear the flag before 10 PCLK clock cycles have elapsed after stopped oscillation was detected.

[Clearing condition]

- Writing 0 to the bit after having read its value as 1.

[Setting condition]

- Detection of the oscillation-stopped state

## 21.3 Operation

The target pins for high-impedance control and conditions to place the pins in high-impedance are described below.

(1) MTU0 pin (MTIOC0A)

When any of the following conditions is satisfied, the pin is placed to the high-impedance state.

- POE8# input level detection
  When ICSR2.POE8F flag is set to 1 with POECR1.PE0ZE and ICSR2.POE8E set to 1.
- SPOER setting
  When SPOER.CH0HIZ bit is set to 1 with POECR1.PE0ZE set to 1.
- Detection of stopped oscillation
  When OSTSTF flag is set to 1 with POECR1.PE0ZE and ICSR3.OSTSTE set to 1.

(2) MTU0 pin (MTIOC0B)

When any of the following conditions is satisfied, the pin is placed to the high-impedance state.

- POE8# input level detection
  When ICSR2.POE8F flag is set to 1 with POECR1.PE1ZE and ICSR2.POE8E set to 1.
- SPOER setting
  When SPOER.CH0HIZ bit is set to 1 with POECR1.PE1ZE set to 1.
- Detection of stopped oscillation
  When OSTSTF flag is set to 1 with POECR1.PE1ZE and ICSR3.OSTSTE set to 1.

(3) MTU0 pin (MTIOC0C)

When any of the following conditions is satisfied, the pin is placed to the high-impedance state.

- POE8# input level detection
  When ICSR2.POE8F flag is set to 1 with POECR1.PE2ZE and ICSR2.POE8E set to 1.
- SPOER setting
  When SPOER.CH0HIZ bit is set to 1 with POECR1.PE2ZE set to 1.
- Detection of stopped oscillation
  When OSTSTF flag is set to 1 with POECR1.PE2ZE and ICSR3.OSTSTE set to 1.

(4) MTU0 pin (MTIOC0D)

When any of the following conditions is satisfied, the pin is placed to the high-impedance state.

- POE8# input level detection
  When ICSR2.POE8F flag is set to 1 with POECR1.PE3ZE and ICSR2.POE8E set to 1.
- SPOER setting
  When SPOER.CH0HIZ bit is set to 1 with POECR1.PE3ZE set to 1.
- Detection of stopped oscillation
  When OSTSTF flag is set to 1 with POECR1.PE3ZE and ICSR3.OSTSTE set to 1.

(5) MTU3 pins (MTIOC3B and MTIOC3D)

When any of the following conditions is satisfied, the pins are placed to the high-impedance state.

- POE0# to POE3# input level detection
  When ICSR1.POE3F, POE2F, POE1F, or POE0F flag is set to 1 with POECR2.P1CZEA set to 1.
- MTIOC3B and MTIOC3D output level comparison
  When OCSR1.OSF1 flag is set to 1 with POECR2.P1CZEA and OCSR1.OCE1 set to 1.
- SPOER setting
  When SPOER.CH34HIZ bit is set to 1 with POECR2.P1CZEA set to 1.
- Detection of stopped oscillation
  When ICSR3.OSTSTF flag is set to 1 with POECR2.P1CZEA and ICSR3.OSTSTE set to 1.

(6) MTU4 pins (MTIOC4A and MTIOC4C)

When any of the following conditions is satisfied, the pins are placed to the high-impedance state.

- POE0# to POE3# input level detection
  When ICSR1.POE3F, POE2F, POE1F, or POE0F flag is set to 1 with POECR2.P2CZEA set to 1.
- MTIOC4A and MTIOC4C output level comparison
  When OCSR1.OSF1 flag is set to 1 with POECR2.P2CZEA and OCSR1.OCE1 set to 1.
- SPOER setting
  When SPOER.CH34HIZ bit is set to 1 with POECR2.P2CZEA set to 1.
- Detection of stopped oscillation
  When ICSR3.OSTSTF flag is set to 1 with POECR2.P2CZEA and ICSR3.OSTSTE set to 1.

(7) MTU4 pins (MTIOC4B and MTIOC4D)

When any of the following conditions is satisfied, the pins are placed to the high-impedance state.

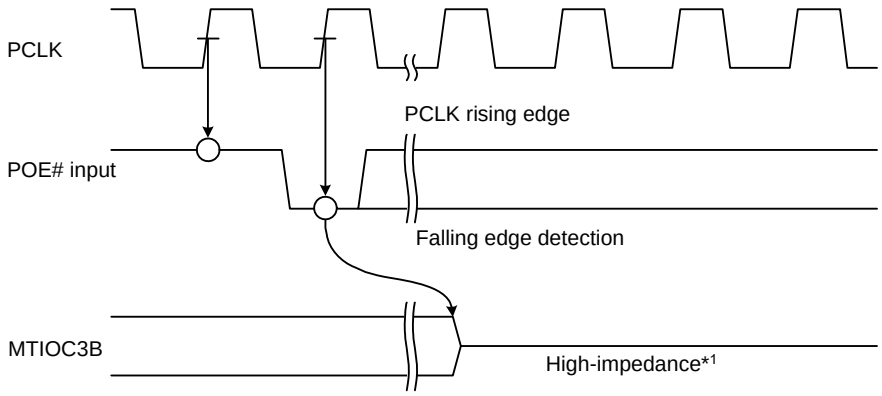
- POE0# to POE3# input level detection
  When ICSR1.POE3F, POE2F, POE1F, or POE0F flag is set to 1 with POECR2.P3CZEA set to 1.
- MTIOC4B and MTIOC4D output level comparison
  When OCSR1.OSF1 flag is set to 1 with POECR2.P3CZEA and OCSR1.OCE1 set to 1.
- SPOER setting
  When SPOER.CH34HIZ bit is set to 1 with POECR2.P3CZEA set to 1.
- Detection of stopped oscillation
  When ICSR3.OSTSTF flag is set to 1 with POECR2.P3CZEA and ICSR3.OSTSTE set to 1.

### 21.3.1 Input Level Detection Operation

If the input conditions set by the ICSR1 and ICSR2 registers occur on the POE0# to POE3# and POE8# pins, the pins for the MTU complementary PWM output and MTU0 are placed in high-impedance.

#### (1) Falling Edge Detection

When a change from a high to low level is input to the POE0# to POE3# and POE8# pins, the pins for the MTU complementary PWM output and MTU0 are placed in high-impedance.
A falling edge is detected after PCLK causes sampling to proceed. If the low level is input to the POE0# to POE3# or POE8# pin over less than one PCLK cycle, whether the falling edge will or will not be detected cannot be guaranteed.
Figure 21.2 shows the timing of sampling after the level changes in input to the POE0# to POE3# and POE8# pins until the respective pins enter high-impedance.

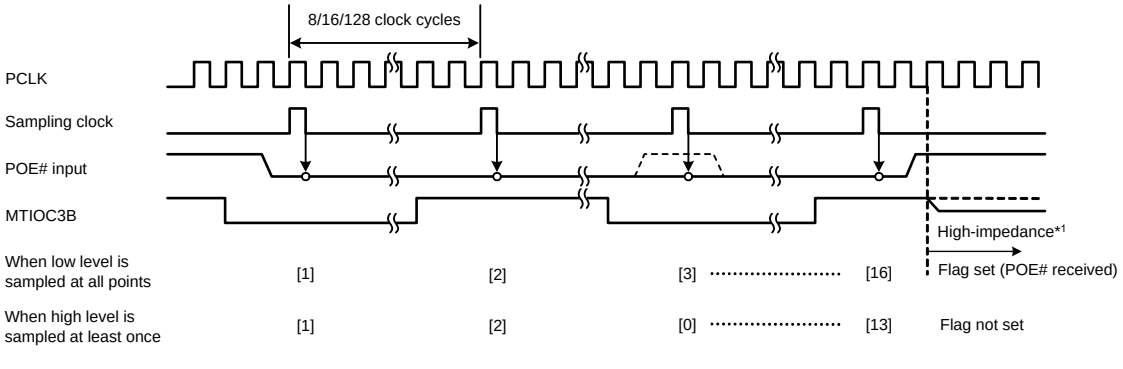



Note 1. The other MTU complementary PWM output pins and MTU0 pins also enter the high-impedance in the similar timing.

**Figure 21.2 Falling Edge Detection**

### (2) Low-Level Detection

Figure 21.3 shows the low-level detection operation. When a low level is detected 16 times continuously with the sampling clock selected by the ICSR1 and ICSR2 registers, the detected level is recognized as low, and the pins for the MTU complementary PWM output and MTU0 are placed in high-impedance. If even one high level is detected during this interval, the detected level is not recognized as low. Furthermore, in an interval over which the sampling clock is not being output, changes to the levels on the POE0# to POE3# and POE8# pins are ignored.

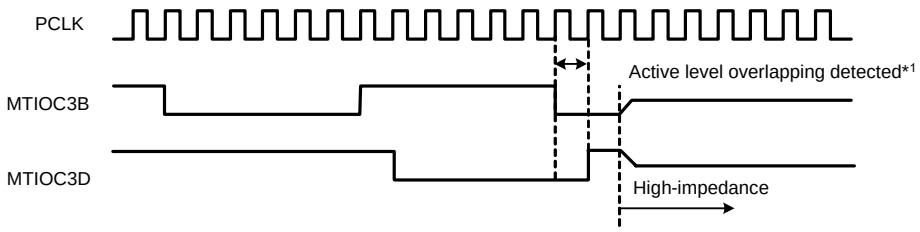



Note 1. The other MTU complementary PWM output pins and MTU0 pins also enter the high-impedance in the similar timing.

**Figure 21.3 Low-Level Detection Operation**

## 21.3.2 Output-Level Compare Operation

Figure 21.4 shows an example of the output-level compare operation for the combination of MTIOC3B and MTIOC3D (MTU complementary PWM output pins). The operation is the same for the other pin combinations.




Note 1. In this case, the low level has been set as the active level for the MTIOC3B and MTIOC3D pins.

**Figure 21.4 Output-Level Compare Operation**

### 21.3.3 High-Impedance Control Using Registers

The high-impedance of the MTU complementary PWM output and MTU0 pins can be directly controlled by writing to the software port output enable register (SPOER).
Setting the SPOER.CH34HIZ bit to 1 places the MTU complementary PWM output pins (MTU3 and MTU4) specified by the POECR2 register in the high-impedance.
Setting the SPOER.CH0HIZ bit to 1 places the MTU0 output pins specified by port output enable control register 1 (POECR1) in the high-impedance.

### 21.3.4 High-Impedance Control on Detection of Stopped Oscillation

When the oscillation stop detection function in the clock generation circuit detects stopped oscillation while the ICSR3.OSTSTE bit is 1, the MTU complementary PWM output pins specified by the POECR2 register and the MTU0 output pins specified by the POECR1 register are placed in the high-impedance.

### 21.3.5 Release from the High-Impedance

Pins for complementary PWM output from MTU and pins for MTU0 which have been placed in the high-impedance due to input-level detection can be released from that state by either returning them to their initial state with a reset or clearing all of the ICSR1.POE3F to POE0F flags and the ICSR2.POE8F flag. Note, however, that when low-level sampling is selected by the ICSR1.POE3M[1:0], POE2M[1:0], POE1M[1:0], and POE0M[1:0] bits, and the ICSR2.POE8M[1:0] bits, if a high level is being input to the corresponding pin from among POE0# to POE3# and POE#8 but has not yet been detected, writing 0 to the flag is ignored (the flag is not cleared).
MTU complementary PWM output pins which have been placed in the high-impedance due to output-level comparison can be released from that state by either returning them to their initial state with a reset or clearing the OCSR1.OSF1 flag. Note, however, that if the inactive level is not yet being output from the MTU complementary PWM output pins, writing 0 to the flag is ignored (the flag is not cleared). Inactive-level outputs can be obtained by setting the MTU registers.
For MTU complementary PWM output pins and pins for MTU0 that have been placed in the high-impedance because oscillation by the clock generation circuit has stopped, clearing the ICSR3.OSTSTF or ICSR3.OSTSTE bit releases the pins from the high-impedance.
For MTU complementary PWM output pins and pins for MTU0 that have been placed in the high-impedance by the SPOER.CH34HIZ or SPOER.CH0HIZ bit, clearing the corresponding bits (SPOER.CH34HIZ and SPOER.CH0HIZ) releases the pins from the high-impedance.

## 21.4 Interrupts

The POE issues a request to generate an interrupt when the corresponding condition below is matched during input-level detection, output-level comparison, or oscillation stop by the clock generation circuit. Table 21.4 lists the interrupt sources and their request conditions. On acceptance of an OEI1 or OEI2 interrupt, the first line of the exception handling routine for the given interrupt should confirm that the flag for the given flag has been set to 1.

**Table 21.4 Interrupt Sources and Conditions**

| Name | Interrupt Source | Interrupt Flag | Condition |
|---|---|---|---|
| OEI1 | Output enable interrupt 1 | POE0F, POE1F, POE2F, POE3F, OSF1 | When ICSR1.POE0F, POE1F, POE2F, or POE3F flag is set to 1 with ICSR1.PIE1 set to 1, or when OCSR1.OSF1 flag is set to 1 with OCSR1.OIE1 set to 1. |
| OEI2 | Output enable interrupt 2 | POE8F | When ICSR2.POE8F flag is set to 1 with ICSR2.PIE2 set to 1. |

## 21.5 Usage Notes

### 21.5.1 Transitions to Software Standby Mode

When the POE is used, do not make a transition to software standby mode. In these modes, the POE stops and thus the high-impedance of pins cannot be controlled.

### 21.5.2 When POE Is Not Used

When the POE is not used, write 00h to port output enable control registers 1 and 2 (POECR1 and POECR2), respectively.

### 21.5.3 Specifying Pins Corresponding to the MTU

The POE controls high-impedance outputs only when a pin has been specified so that the pin corresponds to the MTU by setting the PMR and PmnPFS registers. When the pin has been specified as a general I/O pin, the POE does not control high-impedance outputs.

# 22. Compare Match Timer (CMT)

This MCU has an on-chip compare match timer (CMT) unit (unit 0) consisting of a two-channel 16-bit timer (i.e., a total of two channels). The CMT has a 16-bit counter, and can generate interrupts at set intervals.

## 22.1 Overview

Table 22.1 lists the specifications for the CMT.
Figure 22.1 shows a block diagram of the CMT (unit 0). A two-channel CMT constitutes a unit.

**Table 22.1 Specifications of CMT**

| Item | Description |
|---|---|
| Count clock | • Four frequency dividing clocks<br>One clock from PCLK/8, PCLK/32, PCLK/128, and PCLK/512 can be selected for each channel. |
| Interrupt | A compare match interrupt can be requested for each channel. |
| Event link function (output) | An event signal is output upon a CMT1 compare match. |
| Event link function (input) | Linking to the specified module is possible.<br>Count start, event count, or count restart is possible upon the specified event. |
| Low power consumption facilities | Module stop state can be set. |

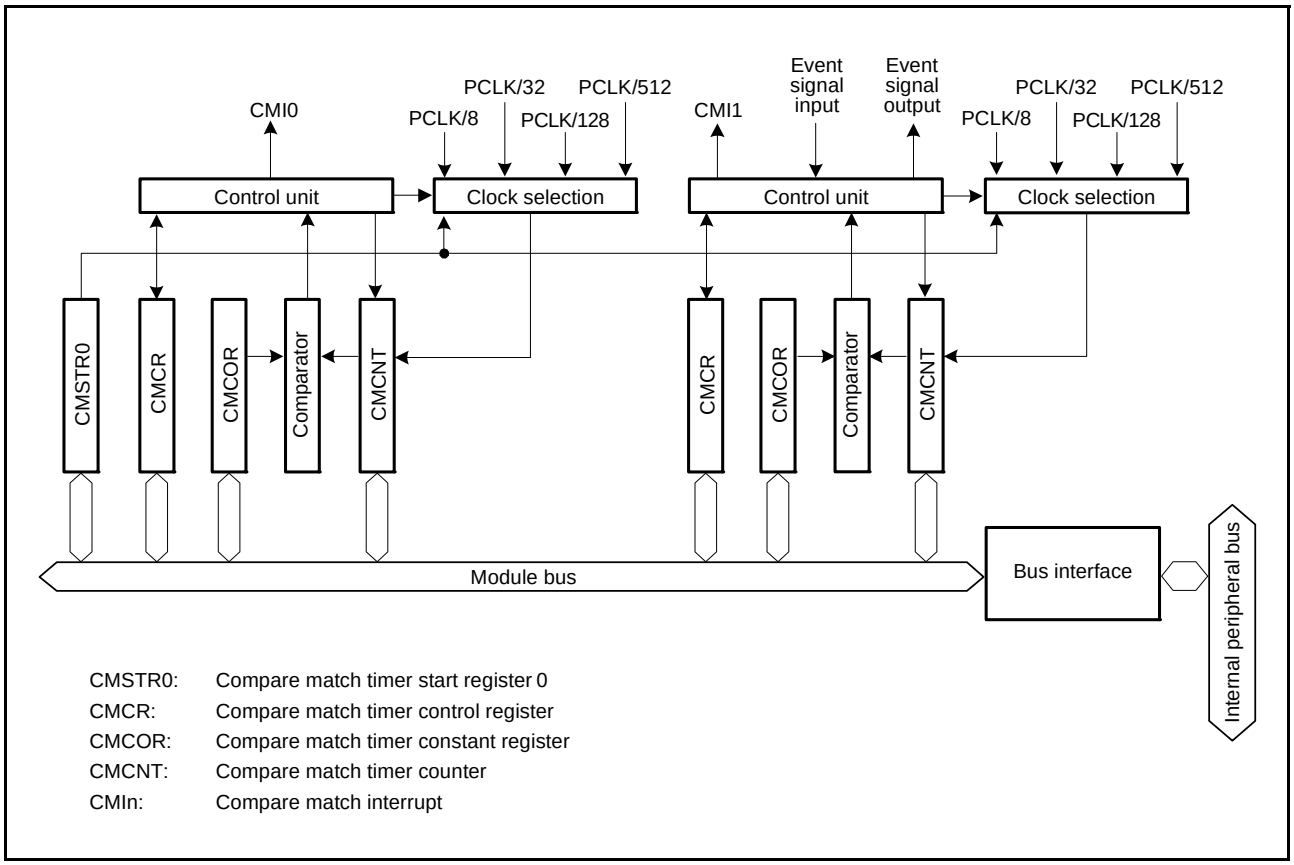



**Figure 22.1 Block Diagram of CMT (Unit 0)**

## 22.2 Register Descriptions

### 22.2.1 Compare Match Timer Start Register 0 (CMSTR0)

Address(es): 0008 8000h

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | STR1 | STR0 |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b0 | STR0 | Count Start 0 | 0: CMT0.CMCNT count is stopped.<br>1: CMT0.CMCNT count is started. | R/W |
| b1 | STR1 | Count Start 1 | 0: CMT1.CMCNT count is stopped.<br>1: CMT1.CMCNT count is started. | R/W |
| b15 to b2 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |

### 22.2.2 Compare Match Timer Control Register (CMCR)

Address(es): CMT0.CMCR 0008 8002h, CMT1.CMCR 0008 8008h

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | CMIE | — | — | — | — | CKS[1:0] | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | x | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

x: Undefined

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b1, b0 | CKS[1:0] | Clock Select | b1 b0<br>0 0: PCLK/8<br>0 1: PCLK/32<br>1 0: PCLK/128<br>1 1: PCLK/512 | R/W |
| b5 to b2 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |
| b6 | CMIE | Compare Match Interrupt Enable | 0: Compare match interrupt (CMIn) disabled<br>1: Compare match interrupt (CMIn) enabled | R/W |
| b7 | — | Reserved | This bit is read as undefined. The write value should be 1. | R/W |
| b15 to b8 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |

#### CKS[1:0] Bits (Clock Select)

These bits select the count source from four frequency dividing clocks obtained by dividing the peripheral module clock (PCLK).

When the CMSTR0.STRn (n = 0, 1) bit is set to 1, the CMCNT counter starts counting up on the clock selected with bits CKS[1:0].

#### CMIE Bit (Compare Match Interrupt Enable)

The CMIE bit enables or disables compare match interrupt (CMIn) (n = 0, 1) generation when the CMCNT counter and the CMCOR register values match.

### 22.2.3 Compare Match Counter (CMCNT)

Address(es): CMT0.CMCNT 0008 8004h, CMT1.CMCNT 0008 800Ah

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

The CMCNT counter is a readable/writable up-counter.
When an frequency dividing clock is selected by the CMCR.CKS[1:0] bits and the CMSTR0.STRn (n = 0, 1) bit is set to 1, the CMCNT counter starts counting up using the selected clock.
When the value in the CMCNT counter and the value in the CMCOR register match, the CMCNT counter is cleared to 0000h. At the same time, a compare match interrupt (CMIn) (n = 0, 1) is generated.

### 22.2.4 Compare Match Constant Register (CMCOR)

Address(es): CMT0.CMCOR 0008 8006h, CMT1.CMCOR 0008 800Ch

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Value after reset: | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

The CMCOR register is a readable/writable register to set a value for compare match with the CMCNT counter.

## 22.3 Operation

### 22.3.1 Periodic Count Operation

When an frequency dividing clock is selected by the CMCR.CKS[1:0] bits and the CMSTR0.STRn (n = 0, 1) bit is set to 1, the CMCNT counter starts counting up using the selected clock.
When the value in the counter and the value in the register match, a compare match interrupt (CMIn) (n = 0,1) is generated. The CMCNT counter then starts counting up again from 0000h. Figure 22.2 shows the operation of the CMCNT counter.

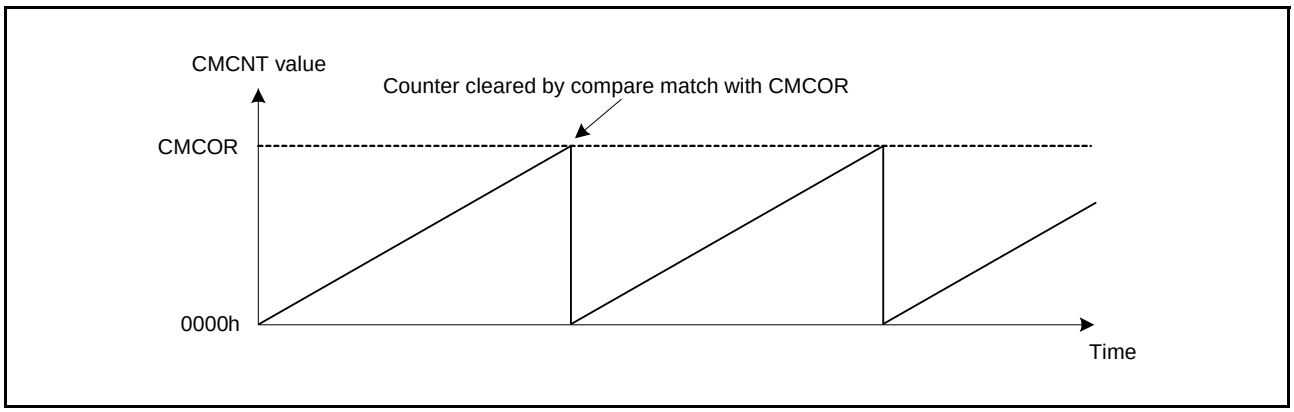



**Figure 22.2 CMCNT Counter Operation**

### 22.3.2 CMCNT Count Timing

As the count clock to be input to the CMCNT counter, one of four frequency dividing clocks (PCLK/8, PCLK/32, PCLK/128, and PCLK/512) obtained by dividing the peripheral clock (PCLK) can be selected with the CMCR.CKS[1:0] bits. Figure 22.3 shows the timing of the CMCNT counter.

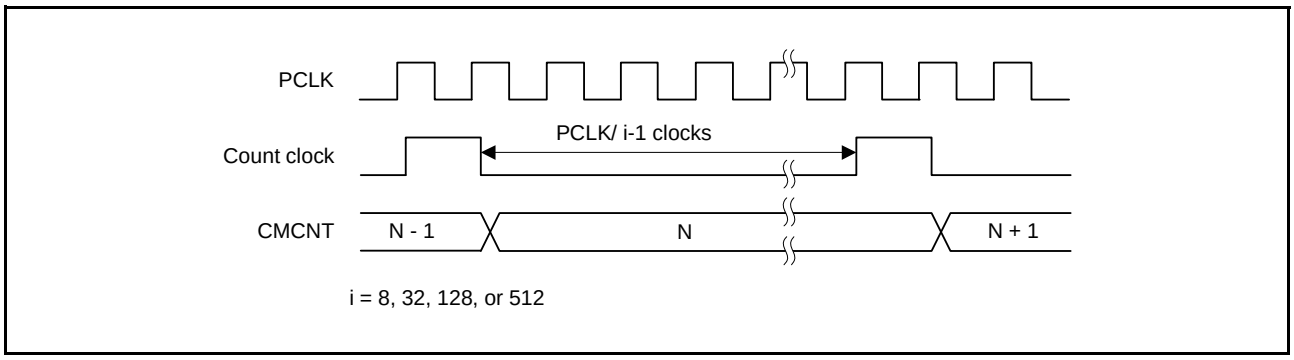



**Figure 22.3 CMCNT Count Timing**

## 22.4 Interrupts

### 22.4.1 Interrupt Sources

The CMT has channels and each of them to which a different vector address is allocated has a compare match interrupt (CMIn) (n = 0, 1). When a compare match interrupt occurs, the corresponding interrupt request is output.
When the interrupt request is used to generate a CPU interrupt, the priority of channels can be changed by the interrupt controller settings. For details, see section 14, Interrupt Controller (ICUb).

**Table 22.2 CMT Interrupt Sources**

| Name | Interrupt Sources | DTC Activation |
|---|---|---|
| CMI0 | Compare match in CMT0 | Possible |
| CMI1 | Compare match in CMT1 | Possible |

### 22.4.2 Timing of Compare Match Interrupt Generation

When the CMCNT counter and the CMCOR register match, a compare match interrupt (CMIn) (n = 0, 1) is generated. A compare match signal is generated at the last state in which the values match (the timing when the CMCNT counter updates the matched count value). That is, after a match between the CMCOR register and the CMCNT counter, the compare match signal is not generated until the next the CMCNT counter input clock.
Figure 22.4 shows the timing of a compare match interrupt.

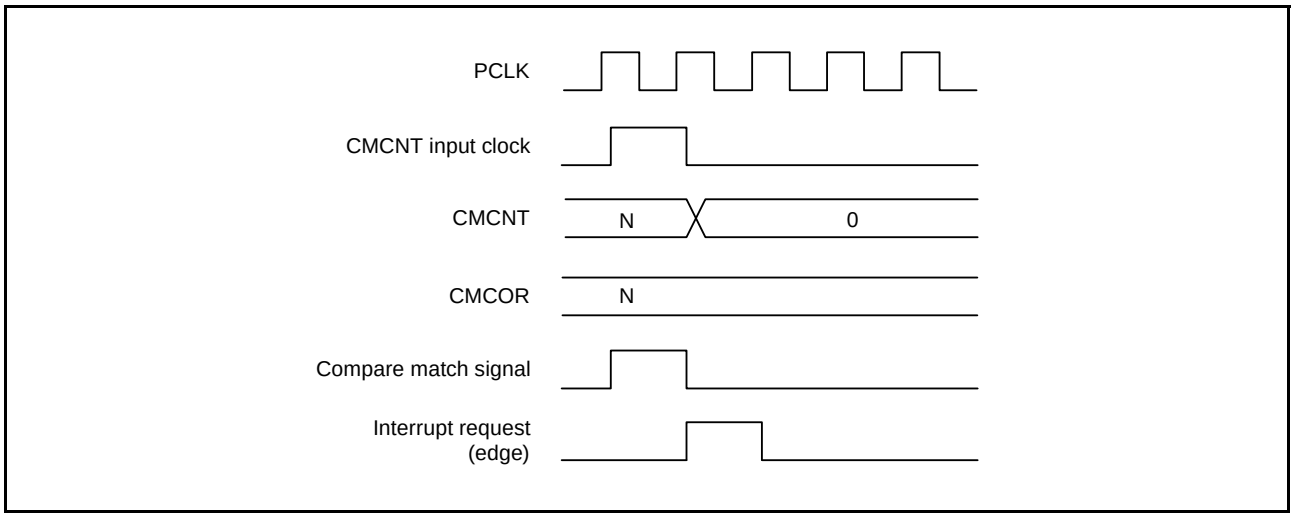



**Figure 22.4 Timing of a Compare Match Interrupt**

## 22.5 Link Operations by ELC

### 22.5.1 Event Signal Output to ELC

The CMT uses the event link controller (ELC) to perform link operation to a preset module using the interrupt request signal as the event signal. The CMT outputs the event signal upon a CMT1 compare match.
The event signal can be output regardless of the setting of the corresponding interrupt request enable bit (CMTn.CMCR.CMIE).

### 22.5.2 CMT Operation when Receiving an Event Signal from ELC

The CMT can perform either of the following operations upon the event preset by the ELSR7 register of the ELC.

#### (1) Count Start

When the CMT count start operation is selected by the ELOPC register of the ELC and the event specified by ELSR7 occurs, the CMSTR0.STR1 bit is set to 1, starting the CMT count operation.
However, if the specified event occurs while the CMSTR0.STR1 bit is 1, the event is ignored.

#### (2) Event Count

When the CMT event count operation is selected by the ELOPC register of the ELC and the event specified by ELSR7 occurs with the CMSTR0.STR1 bit being 1, the events are counted as the count source regardless of the CMT1.CMCR.CKS[1:0] bit setting. Reading the counter value returns the number of events that have been actually input.

#### (3) Count Restart

When the CMT count restart operation is selected by the ELOPC register of the ELC and the event specified by ELSR7 occurs, the CMT1.CMCNT counter value is modified to the initial value. If the CMSTR0.STR1 bit is 1 here, the count operation can be continued.

### 22.5.3 Notes on Operating CMT According to an Event Signal from ELC

The following describes the notes on operating the CMT using the event link feature.

#### (1) Count Start

When the event specified by ELSR7 occurs during the write cycle to the CMSTR0.STR1 bit, the cycle is not completed; setting 1 according to the event occurrence takes priority.

#### (2) Event Count

When the event specified by ELSR7 occurs during the write cycle to the CMT1.CMCNT, the cycle is not completed; event count operation according to the event occurrence takes priority.

#### (3) Count Restart

When the event specified by ELSR7 occurs during the write cycle to the CMT1.CMCNT, the cycle is not completed; count value initialization according to the event occurrence takes priority.

## 22.6 Usage Notes

### 22.6.1 Setting the Module Stop Function

The CMT can be enabled or disabled using the module stop control register. After a reset, the CMT is in the module stop state. The registers can be accessed by canceling the module stop state. For details, see section 11, Low Power Consumption.

### 22.6.2 Conflict between Write and Compare-Match Processes of CMCNT

When the compare match signal is generated while writing to the CMCNT counter, clearing the CMCNT counter has priority over writing to it. In this case, the CMCNT counter is not written to. Figure 22.5 shows the timing to clear the CMCNT counter.

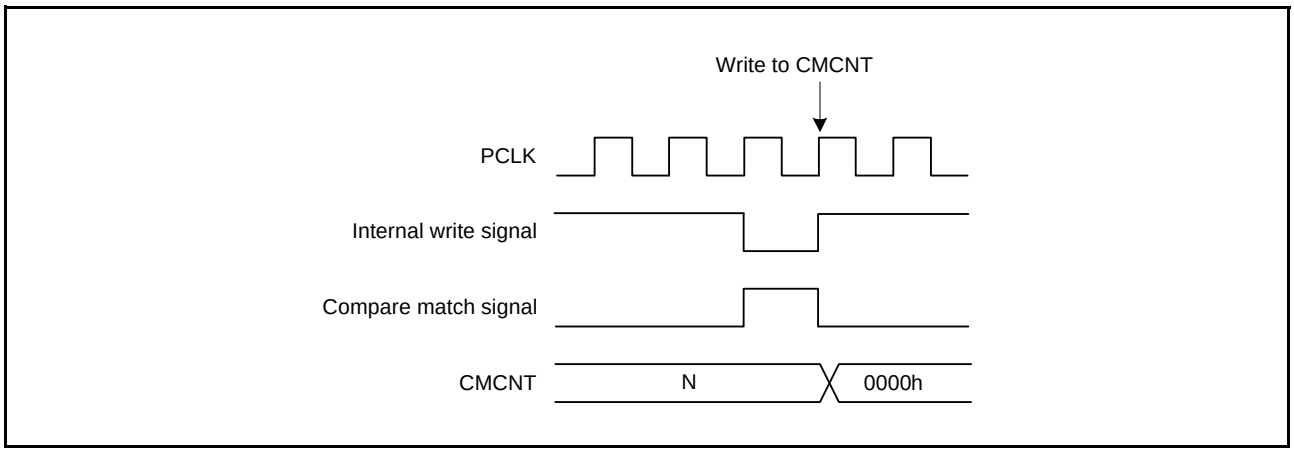



**Figure 22.5 Conflict between Write and Compare Match Processes of CMCNT**

### 22.6.3 Conflict between Write and Count-Up Processes of CMCNT

If writing to the counter and the count-up conflict, the writing has priority over the count-up. Figure 22.6 shows the timing to write the CMCNT counter.

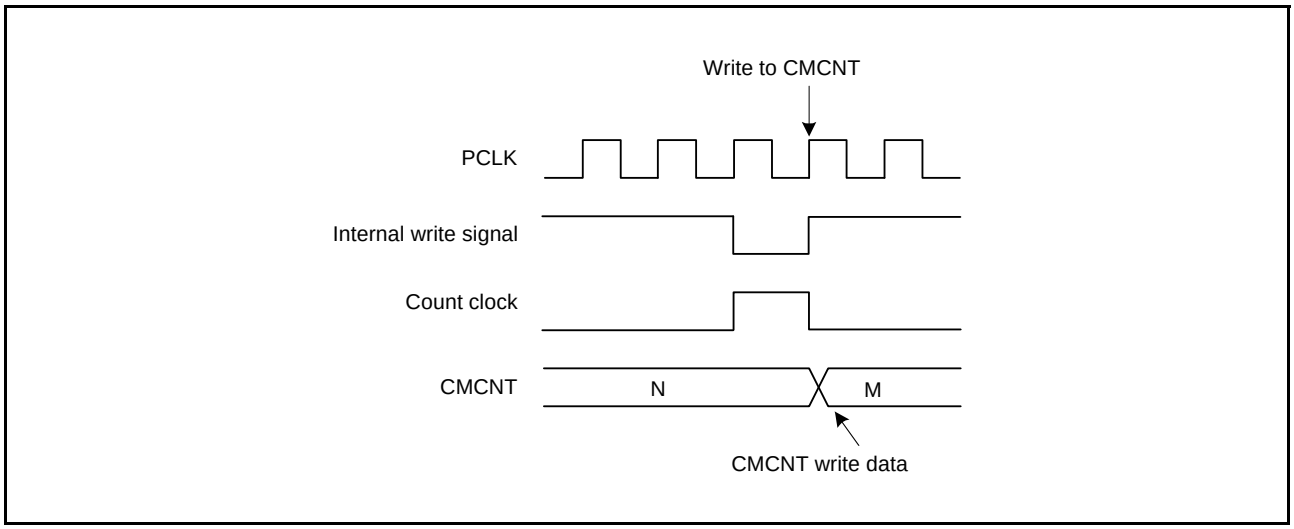



**Figure 22.6 Conflict between Write and Count-Up Processes of CMCNT**

# 23. Realtime Clock (RTCA)

## 23.1 Overview

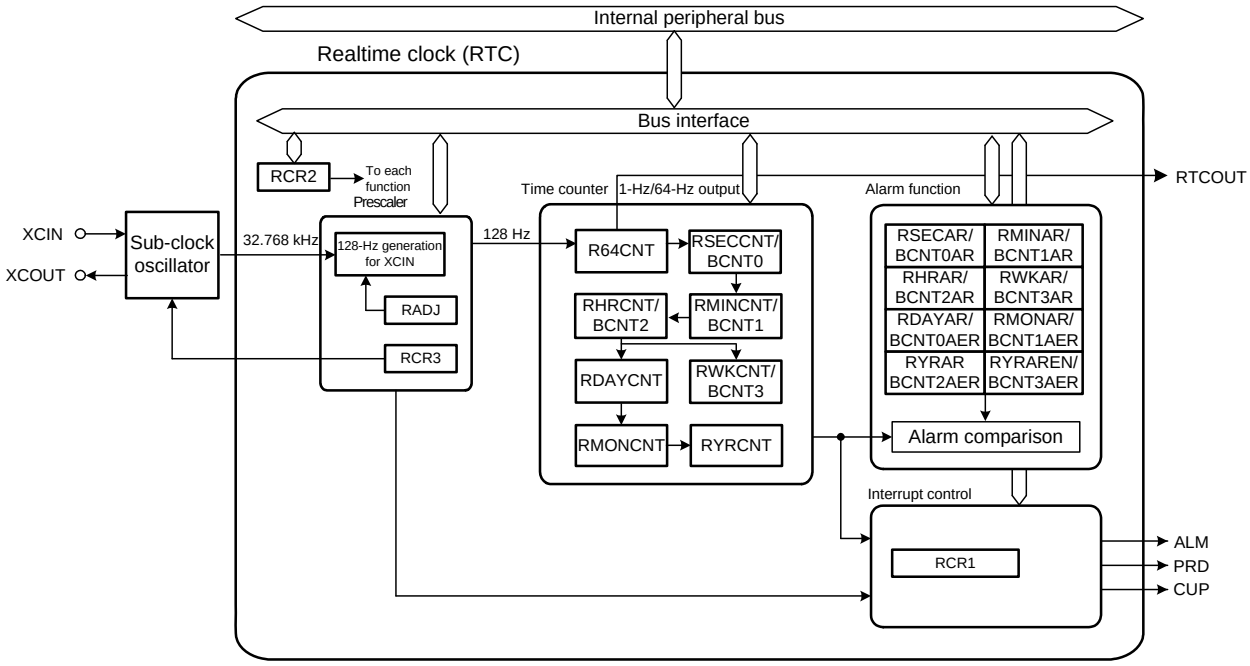
The RTC has two types of counting modes: calendar count mode and binary count mode. They are used by switching the register settings.
For calendar count mode, the RTC has a 100 year calendar from 2000 to 2099 and automatically adjusts dates for leap years.
For binary count mode, the RTC does not count in terms of years, months, dates, day-of-week, hours, or minutes; it counts seconds, and retains the information as a serial value. This mode can be used for calendars other than the Gregorian calendar.
The RTC uses the 128-Hz clock which is acquired by the count source divided by the prescaler as the basic clock. Year, month, date, day-of-week, a.m./p.m. (in 12-hour mode), hour, minute, second, or 32-bit binary is counted in 1/128 second units.

Table 23.1 lists the specifications of the RTC, Figure 23.1 shows a block diagram of the RTC, and Table 23.2 shows the pin configuration of the RTC.

**Table 23.1 RTC Specifications**

| Item | Description |
|---|---|
| Count mode | Calendar count mode/binary count mode |
| Count source*1 | Sub-clock (XCIN) |
| Clock and calendar functions | • Calendar count mode<br>Year, month, date, day-of-week, hour, minute, second are counted, BCD display<br>12 hours/24 hours mode switching function<br>30 seconds adjustment function (a number less than 30 is rounded down to 00 seconds, and 30 seconds or more are rounded up to one minute)<br>Automatic adjustment function for leap years<br>• Binary count mode<br>Count seconds in 32 bits, binary display<br>• Common to both modes<br>Start/stop function<br>The sub-second digit is displayed in binary units (1 Hz, 2 Hz, 4 Hz, 8 Hz, 16 Hz, 32 Hz, or 64 Hz).<br>Clock error correction function<br>Clock (1-Hz/64-Hz) output |
| Interrupts | • Alarm interrupt (ALM)<br>As an alarm interrupt condition, selectable which of the below is compared with:<br>Calendar count mode: Year, month, date, day-of-week, hour, minute, or second can be selected<br>Binary count mode: Each bit of the 32-bit binary counter<br>• Periodic interrupt (PRD)<br>2 seconds, 1 second, 1/2 second, 1/4 second, 1/8 second, 1/16 second, 1/32 second, 1/64 second, or 1/256 second can be selected as an interrupt period.<br>• Carry interrupt (CUP)<br>An interrupt is generated at either of the following timings:<br>- When a carry from the 64-Hz counter to the second counter is generated.<br>- When the 64-Hz counter is changed and the R64CNT register is read at the same time.<br>• Recovery from software standby mode can be performed by an alarm interrupt or periodic interrupt |

Note 1. Satisfy the frequency of the peripheral module clock (PCLKB) ≥ the frequency of the count source clock.




R64CNT: 64-Hz counter
RSECCNT/BCNT0: Second counter/Binary counter 0
RMINCNT/BCNT1: Minute counter/Binary counter 1
RHRCNT/BCNT2: Hour counter/Binary counter 2
RWKCNT/BCNT3: Day-of-week counter/Binary counter 3
RDAYCNT: Date counter
RMONCNT: Month counter
RYRCNT: Year counter
RCR1: RTC control register 1
RCR2: RTC control register 2
RCR3: RTC control register 3
RADJ: Time error adjustment register

RSECAR/BCNT0AR: Second alarm register/Binary counter 0 alarm register
RMINAR/BCNT1AR: Minute alarm register/Binary counter 1 alarm register
RHRAR/BCNT2AR: Hour alarm register/Binary counter 2 alarm register
RWKAR/BCNT3AR: Day-of-week alarm register/Binary counter 3 alarm register
RDAYAR/BCNT0AER: Date alarm register/Binary counter 0 alarm enable register
RMONAR/BCNT1AER: Month alarm register/Binary counter 1 alarm enable register
RYRAR/BCNT2AER: Year alarm register/Binary counter 2 alarm enable register
RYRAREN/BCNT3AER: Year alarm enable register/Binary counter 3 alarm enable register

**Figure 23.1 Block Diagram of RTC**

**Table 23.2 Pin Configuration of RTC**

| Pin Name | I/O | Function |
|---|---|---|
| XCIN | Input | Connect a 32.768-kHz crystal to these pins. |
| XCOUT | Output | |
| RTCOUT | Output | Output a 1-Hz/64-Hz clock. |

## 23.2 Register Descriptions

When writing to or reading from RTC registers, do so in accord with section 23.5.5, Points for Caution when Writing to and Reading from Registers.

If the value in an RTC register after a reset is given as x (undefined bits) in the list, it is not initialized by a reset. When RTC enters the reset state or a low power consumption state during counting operations (i.e. while the RCR2.START bit is 1), the year, month, day of the week, date, hours, minutes, seconds, and 64-Hz counters continue to operate. However, note that a reset generated during writing to or updating of a register might destroy the register value. In addition, do not allow the chip to enter software standby mode immediately after setting any of these registers. For details, refer to section 23.5.4, Transitions to Low Power Consumption Modes after Setting Registers.

### 23.2.1 64-Hz Counter (R64CNT)

Address(es): 0008 C400h

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | F1HZ | F2HZ | F4HZ | F8HZ | F16HZ | F32HZ | F64HZ |
| Value after reset: | 0 | x | x | x | x | x | x | x |

x: Undefined

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b0 | F64HZ | 64 Hz | Indicate the state between 1 Hz and 64 Hz of the sub-second digit. | R |
| b1 | F32HZ | 32 Hz | | R |
| b2 | F16HZ | 16 Hz | | R |
| b3 | F8HZ | 8 Hz | | R |
| b4 | F4HZ | 4 Hz | | R |
| b5 | F2HZ | 2 Hz | | R |
| b6 | F1HZ | 1 Hz | | R |
| b7 | — | Reserved | This bit is read as 0. Writing to this bit has no effect. | R |

The R64CNT counter is used in both calendar count mode and in binary count mode.

The 64-Hz counter (R64CNT) generates the period for a second by counting up periods of the 128-Hz clock.

The state in the sub-second range can be confirmed by reading this counter.

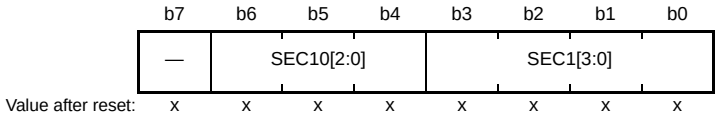
This counter is cleared to 00h by an RTC software reset or executing 30-second adjustment.

To read this counter, follow the procedure in section 23.3.5, Reading 64-Hz Counter and Time.

## 23.2.2 Second Counter (RSECCNT)/Binary Counter 0 (BCNT0)

### (1) In calendar count mode:

Address(es): 0008 C402h



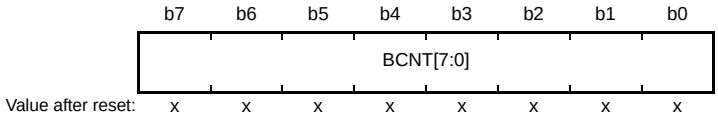

x: Undefined

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b3 to b0 | SEC1[3:0] | 1-Second Count | Counts from 0 to 9 every second. When a carry is generated, 1 is added to the tens place. | R/W |
| b6 to b4 | SEC10[2:0] | 10-Second Count | Counts from 0 to 5 for 60-second counting. | R/W |
| b7 | — | Reserved | Set this bit to 0. It is read as the set value. | R/W |

The RSECCNT counter is used for setting and counting the BCD-coded second value. It counts carries generated once per second in the 64-Hz counter.
The setting range is decimal 00 to 59. The RTC will not operate normally if any other value is set. Before writing to this register, be sure to stop the count operation through the setting of the START bit in RCR2.
To read this counter, follow the procedure in section 23.3.5, Reading 64-Hz Counter and Time.

### (2) In binary count mode:

Address(es): 0008 C402h



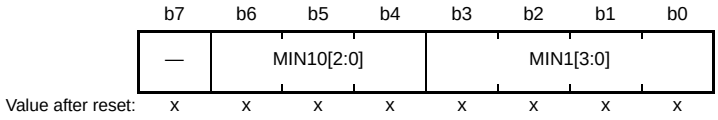

x: Undefined

The BCNT0 counter is a readable/writable 32-bit binary counter b7 to b0.
The 32-bit binary counter performs count operation by a carry generating for each second of the 64-Hz counter.
Before writing to this register, be sure to stop the count operation through the setting of the START bit in RCR2.
To read this counter, follow the procedure in section 23.3.5, Reading 64-Hz Counter and Time.

### 23.2.3 Minute Counter (RMINCNT)/Binary Counter 1 (BCNT1)

#### (1) In calendar count mode:

Address(es): 0008 C404h



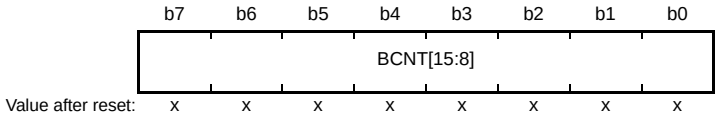

x: Undefined

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b3 to b0 | MIN1[3:0] | 1-Minute Count | Counts from 0 to 9 every minute. When a carry is generated, 1 is added to the tens place. | R/W |
| b6 to b4 | MIN10[2:0] | 10-Minute Count | Counts from 0 to 5 for 60-minute counting. | R/W |
| b7 | — | Reserved | Set this bit to 0. It is read as the set value. | R/W |

The RMINCNT counter is used for setting and counting the BCD-coded minute value. It counts carries generated once per minute in the second counter.
A value from 00 through 59 (in BCD) can be specified; if a value outside of this range is specified, the RTC does not operate correctly. Before writing to this register, be sure to stop the count operation through the setting of the START bit in RCR2.
To read this counter, follow the procedure in section 23.3.5, Reading 64-Hz Counter and Time.

#### (2) In binary count mode:

Address(es): 0008 C404h



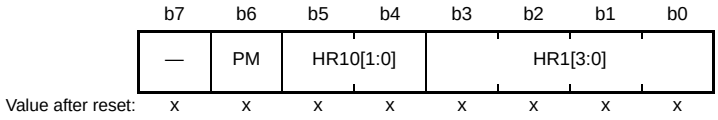

x: Undefined

The BCNT1 counter is a readable/writable 32-bit binary counter b15 to b8.
The 32-bit binary counter performs count operation by a carry generating for each second of the 64-Hz counter.
Before writing to this register, be sure to stop the count operation through the setting of the START bit in RCR2.
To read this counter, follow the procedure in section 23.3.5, Reading 64-Hz Counter and Time.

### 23.2.4 Hour Counter (RHRCNT)/Binary Counter 2 (BCNT2)

#### (1) In calendar count mode:

Address(es): 0008 C406h



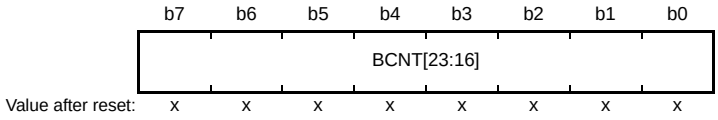

x: Undefined

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b3 to b0 | HR1[3:0] | 1-Hour Count | Counts from 0 to 9 once per hour. When a carry is generated, 1 is added to the tens place. | R/W |
| b5, b4 | HR10[1:0] | 10-Hour Count | Counts from 0 to 2 once per carry from the ones place. | R/W |
| b6 | PM | PM | Time Counter Setting for a.m./p.m.<br>0: a.m.<br>1: p.m. | R/W |
| b7 | — | Reserved | Set this bit to 0. It is read as the set value. | R/W |

The RHRCNT counter is used for setting and counting the BCD-coded hour value. It counts carries generated once per hour in the minute counter.
The specifiable time differs according to the setting in the hours mode bit (RCR2.HR24).
When the RCR2.HR24 bit is 0: From 00 to 11 (in BCD)
When the RCR2.HR24 bit is 1: From 00 to 23 (in BCD)
If a value outside of this range is specified, the RTC does not operate correctly. Before writing to this register, be sure to stop the count operation through the setting of the START bit in RCR2.
The PM bit is only enabled when the RCR2.HR24 bit is 0. Otherwise, the setting in the PM bit has no effect.
To read this counter, follow the procedure in section 23.3.5, Reading 64-Hz Counter and Time.

#### (2) In binary count mode:

Address(es): 0008 C406h




x: Undefined

The BCNT2 counter is a readable/writable 32-bit binary counter b23 to b16.
The 32-bit binary counter performs count operation by a carry generating for each second of the 64-Hz counter.
Before writing to this register, be sure to stop the count operation through the setting of the START bit in RCR2.
To read this counter, follow the procedure in section 23.3.5, Reading 64-Hz Counter and Time.

## 23.2.5 Day-of-Week Counter (RWKCNT)/Binary Counter 3 (BCNT3)

### (1) In calendar count mode:

Address(es): 0008 C408h

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | DAYW[2:0] | | |
| Value after reset: | x | x | x | x | x | x | x | x |

x: Undefined

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b2 to b0 | DAYW[2:0] | Day-of-Week Counting | b2 b0<br>0 0 0: Sunday<br>0 0 1: Monday<br>0 1 0: Tuesday<br>0 1 1: Wednesday<br>1 0 0: Thursday<br>1 0 1: Friday<br>1 1 0: Saturday<br>1 1 1: Setting Prohibited | R/W |
| b7 to b3 | — | Reserved | Set these bits to 0. It is read as the set value. | R/W |

The RWKCNT counter is used for setting and counting in the coded day-of-week value. It counts carries generated once per day in the hour counter.
A value from 0 through 6 can be specified; if a value outside of this range is specified, the RTC does not operate correctly. Before writing to this register, be sure to stop the count operation through the setting of the START bit in RCR2.
To read this counter, follow the procedure in section 23.3.5, Reading 64-Hz Counter and Time.

### (2) In binary count mode:

Address(es): 0008 C408h

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | BCNT[31:24] | | | | | | | |
| Value after reset: | x | x | x | x | x | x | x | x |

x: Undefined

The BCNT3 counter is a readable/writable 32-bit binary counter b31 to b24.
The 32-bit binary counter performs count operation by a carry generating for each second of the 64-Hz counter.
Before writing to this register, be sure to stop the count operation through the setting of the START bit in RCR2.
To read this counter, follow the procedure in section 23.3.5, Reading 64-Hz Counter and Time.

## 23.2.6 Date Counter (RDAYCNT)

Address(es): 0008 C40Ah

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | DATE10[1:0] | | DATE1[3:0] | | | |
| Value after reset: | 0 | 0 | x | x | x | x | x | x |

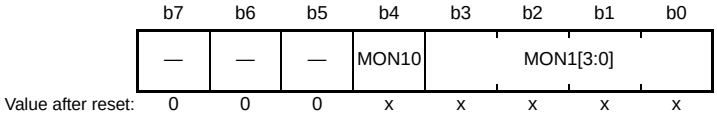
x: Undefined

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b3 to b0 | DATE1[3:0] | 1-Day Count | Counts from 0 to 9 once per day. When a carry is generated, 1 is added to the tens place. | R/W |
| b5, b4 | DATE10[1:0] | 10-Day Count | Counts from 0 to 3 once per carry from the ones place. | R/W |
| b7, b6 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |

The RDAYCNT counter is used in calendar count mode.
RDAYCNT is used for setting and counting the BCD-coded date value. It counts carries generated once per day in the hour counter. The count operation depends on the month and whether the year is a leap year.
Leap years are determined according to whether the year counter (RYRCNT) value is divisible by 400, 100, and 4.
A value from 01 through 31 (in BCD) can be specified; if a value outside of this range is specified, the RTC does not operate correctly. (When specifying a value, note that the range of specifiable days depends on the month and whether the year is a leap year.) Before writing to this register, be sure to stop the count operation through the setting of the START bit in RCR2.
To read this counter, follow the procedure in section 23.3.5, Reading 64-Hz Counter and Time.

## 23.2.7 Month Counter (RMONCNT)

Address(es): 0008 C40Ch



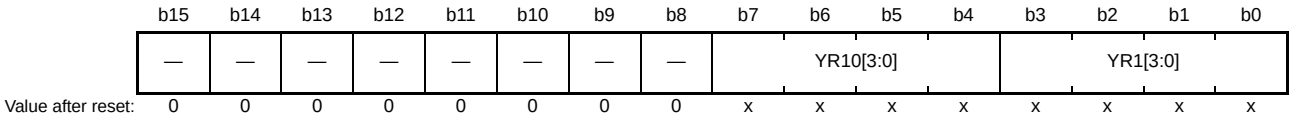

x: Undefined

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b3 to b0 | MON1[3:0] | 1-Month Count | Counts from 0 to 9 once per month. When a carry is generated, 1 is added to the tens place. | R/W |
| b4 | MON10 | 10-Month Count | Counts from 0 to 1 once per carry from the ones place. | R/W |
| b7 to b5 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |

The RMONCNT counter is used in calendar count mode.
RMONCNT is used for setting and counting the BCD-coded month value. It counts carries generated once per month in the date counter.
A value from 01 through 12 (in BCD) can be specified; if a value outside of this range is specified, the RTC does not operate correctly. Before writing to this register, be sure to stop the count operation through the setting of the START bit in RCR2.
To read this counter, follow the procedure in section 23.3.5, Reading 64-Hz Counter and Time.

## 23.2.8 Year Counter (RYRCNT)

Address(es): 0008 C40Eh




x: Undefined

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b3 to b0 | YR1[3:0] | 1-Year Count | Counts from 0 to 9 once per year. When a carry is generated, 1 is added to the tens place. | R/W |
| b7 to b4 | YR10[3:0] | 10-Year Count | Counts from 0 to 9 once per carry from ones place. When a carry is generated in the tens place, 1 is added to the hundreds place. | R/W |
| b15 to b8 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |

The RYRCNT counter is used in calendar count mode.
RYRCNT is used for setting and counting the BCD-coded year value. It counts carries generated once per year in the month counter.
A value from 00 through 99 (in BCD) can be specified; if a value outside of this range is specified, the RTC does not operate correctly. Before writing to this register, be sure to stop the count operation through the setting of the START bit in RCR2.
To read this counter, follow the procedure in section 23.3.5, Reading 64-Hz Counter and Time.

## 23.2.9 Second Alarm Register (RSECAR)/Binary Counter 0 Alarm Register (BCNT0AR)

### (1) In calendar count mode:

Address(es): 0008 C410h

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | ENB | SEC10[2:0] | | | SEC1[3:0] | | | |
| Value after reset: | x | x | x | x | x | x | x | x |

x: Undefined

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b3 to b0 | SEC1[3:0] | 1 Second | Value for the ones place of seconds | R/W |
| b6 to b4 | SEC10[2:0] | 10 Seconds | Value for the tens place of seconds | R/W |
| b7 | ENB | ENB | 0: The register value is not compared with the RSECCNT counter value<br>1: The register value is compared with the RSECCNT counter value | R/W |

RSECAR is an alarm register corresponding to the BCD-coded second counter RSECCNT. When the ENB bit is set to 1, the RSECAR value is compared with the RSECCNT value. From among the alarm registers (RSECAR, RMINAR, RHRAR, RWKAR, RDAYAR, RMONAR, and RYRAREN), only those selected with the ENB bits set to 1 are compared with the corresponding counters. When the respective values all match, the IR flag corresponding to the ALM interrupt is set to 1.
RSECAR values from 00 through 59 (in BCD) can be specified; if a value outside of this range is specified, the RTC does not operate correctly.
This register is cleared to 00h by an RTC software reset.

### (2) In binary count mode:

Address(es): 0008 C410h

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | BCNTAR[7:0] | | | | | | | |
| Value after reset: | x | x | x | x | x | x | x | x |

x: Undefined

The BCNT0AR counter is a readable/writable alarm register corresponding to 32-bit binary counter b7 to b0.
This register is cleared to 00h by an RTC software reset.

## 23.2.10 Minute Alarm Register (RMINAR)/Binary Counter 1 Alarm Register (BCNT1AR)

### (1) In calendar count mode:

Address(es): 0008 C412h

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | ENB | MIN10[2:0] | | | MIN1[3:0] | | | |
| Value after reset: | x | x | x | x | x | x | x | x |

x: Undefined

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b3 to b0 | MIN1[3:0] | 1 Minute | Value for the ones place of minutes | R/W |
| b6 to b4 | MIN10[2:0] | 10 Minutes | Value for the tens place of minutes | R/W |
| b7 | ENB | ENB | 0: The register value is not compared with the RMINCNT counter value<br>1: The register value is compared with the RMINCNT counter value | R/W |

RMINAR is an alarm register corresponding to the BCD-coded minute counter RMINCNT. When the ENB bit is set to 1, the RMINAR value is compared with the RMINCNT value. From among the alarm registers (RSECAR, RMINAR, RHRAR, RWKAR, RDAYAR, RMONAR, and RYRAREN), only those selected with the ENB bits set to 1 are compared with the corresponding counters. When the respective values all match, the IR flag corresponding to the ALM interrupt is set to 1.
RMINAR values from 00 through 59 (in BCD) can be specified; if a value outside of this range is specified, the RTC does not operate correctly.
This register is cleared to 00h by an RTC software reset.

### (2) In binary count mode:

Address(es): 0008 C412h

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | BCNTAR[15:8] | | | | | | | |
| Value after reset: | x | x | x | x | x | x | x | x |

x: Undefined

The BCNT1AR counter is a readable/writable alarm register corresponding to 32-bit binary counter b15 to b8.
This register is cleared to 00h by an RTC software reset.

## 23.2.11 Hour Alarm Register (RHRAR)/Binary Counter 2 Alarm Register (BCNT2AR)

### (1) In calendar count mode:

Address(es): 0008 C414h

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | ENB | PM | HR10[1:0] | | HR1[3:0] | | | |
| Value after reset: | x | x | x | x | x | x | x | x |

x: Undefined

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b3 to b0 | HR1[3:0] | 1 Hour | Value for the ones place of hours | R/W |
| b5, b4 | HR10[1:0] | 10 Hours | Value for the tens place of hours | R/W |
| b6 | PM | PM | Time Alarm Setting for a.m./p.m.<br>0: a.m.<br>1: p.m. | R/W |
| b7 | ENB | ENB | 0: The register value is not compared with the RHRCNT counter value<br>1: The register value is compared with the RHRCNT counter value | R/W |

RHRAR is an alarm register corresponding to the BCD-coded hour counter RHRCNT. When the ENB bit is set to 1, the RHRAR value is compared with the RHRCNT value. From among the alarm registers (RSECAR, RMINAR, RHRAR, RWKAR, RDAYAR, RMONAR, and RYRAREN), only those selected with the ENB bits set to 1 are compared with the corresponding counters. When the respective values all match, the IR flag corresponding to the ALM interrupt is set to 1.
The specifiable time differs according to the setting in the hours mode bit (RCR2.HR24).
When the RCR2.HR24 bit is 0: From 00 to 11 (in BCD)
When the RCR2.HR24 bit is 1: From 00 to 23 (in BCD)
If a value outside of this range is specified, the RTC does not operate correctly.
When the RCR2.HR24 bit is 0, be sure to set the PM bit.
When the RCR2.HR24 bit is 1, the setting in the PM bit has no effect.
This register is cleared to 00h by an RTC software reset.

### (2) In binary count mode:

Address(es): 0008 C414h

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | BCNTAR[23:16] | | | | | | | |
| Value after reset: | x | x | x | x | x | x | x | x |

x: Undefined

The BCNT2AR counter is a readable/writable alarm register corresponding to 32-bit binary counter b23 to b16.
This register is cleared to 00h by an RTC software reset.

## 23.2.12 Day-of-Week Alarm Register (RWKAR)/Binary Counter 3 Alarm Register (BCNT3AR)

### (1) In calendar count mode:

Address(es): 0008 C416h

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | ENB | — | — | — | — | DAYW[2:0] | | |
| Value after reset: | x | x | x | x | x | x | x | x |

x: Undefined

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b2 to b0 | DAYW[2:0] | Day-of-Week Setting | b2 b0<br>0 0 0: Sunday<br>0 0 1: Monday<br>0 1 0: Tuesday<br>0 1 1: Wednesday<br>1 0 0: Thursday<br>1 0 1: Friday<br>1 1 0: Saturday<br>1 1 1: Setting Prohibited | R/W |
| b6 to b3 | — | Reserved | Set these bits to 0. It is read as the set value. | R/W |
| b7 | ENB | ENB | 0: The register value is not compared with the RWKCNT counter value<br>1: The register value is compared with the RWKCNT counter value | R/W |

RWKAR is an alarm register corresponding to the coded day-of-week counter RWKCNT. When the ENB bit is set to 1, the RWKAR value is compared with the RWKCNT value. From among the alarm registers (RSECAR, RMINAR, RHRAR, RWKAR, RDAYAR, RMONAR, and RYRAREN), only those selected with the ENB bits set to 1 are compared with the corresponding counters. When the respective values all match, the IR flag corresponding to the ALM interrupt is set to 1.
RWKAR values from 0 through 6 (in BCD) can be specified; if a value outside of this range is specified, the RTC does not operate correctly.
This register is cleared to 00h by an RTC software reset.

### (2) In binary count mode:

Address(es): 0008 C416h

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | BCNTAR[31:24] | | | | | | | |
| Value after reset: | x | x | x | x | x | x | x | x |

x: Undefined

The BCNT3AR counter is a readable/writable alarm register corresponding to 32-bit binary counter b31 to b24.
This register is cleared to 00h by an RTC software reset.

## 23.2.13 Date Alarm Register (RDAYAR)/Binary Counter 0 Alarm Enable Register (BCNT0AER)

### (1) In calendar count mode:

Address(es): 0008 C418h

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | ENB | — | DATE10[1:0] | | DATE1[3:0] | | | |
| Value after reset: | x | x | x | x | x | x | x | x |

x: Undefined

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b3 to b0 | DATE1[3:0] | 1 Day | Value for the ones place of days | R/W |
| b5, b4 | DATE10[1:0] | 10 Days | Value for the tens place of days | R/W |
| b6 | — | Reserved | Set this bit to 0. It is read as the set value. | R/W |
| b7 | ENB | ENB | 0: The register value is not compared with the RDAYCNT counter value<br>1: The register value is compared with the RDAYCNT counter value | R/W |

RDAYAR is an alarm register corresponding to the BCD-coded date counter RDAYCNT. When the ENB bit is set to 1, the RDAYAR value is compared with the RDAYCNT value. From among the alarm registers (RSECAR, RMINAR, RHRAR, RWKAR, RDAYAR, RMONAR, and RYRAREN), only those selected with the ENB bits set to 1 are compared with the corresponding counters. When the respective values all match, the IR flag corresponding to the ALM interrupt is set to 1.
RDAYAR values from 01 through 31 (in BCD) can be specified; if a value outside of this range is specified, the RTC does not operate correctly.
This register is cleared to 00h by an RTC software reset.

### (2) In binary count mode:

Address(es): 0008 C418h

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | ENB[7:0] | | | | | | | |
| Value after reset: | x | x | x | x | x | x | x | x |

x: Undefined

The BCNT0AER register is a readable/writable register for setting the alarm enable corresponding to 32-bit binary counter b7 to b0. Among the ENB[31:0] bits, the binary counter (BCNT[31:0]) corresponding to the bits which are set to 1 and the binary alarm register (BCNTAR[31:0]) are compared, and when all match, the IR flag corresponding to the ALM interrupt becomes 1.
This register is cleared to 00h by an RTC software reset.

### 23.2.14 Month Alarm Register (RMONAR)/Binary Counter 1 Alarm Enable Register (BCNT1AER)

#### (1) In calendar count mode:

Address(es): 0008 C41Ah

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | ENB | — | — | MON10 | MON1[3:0] | | | |
| Value after reset: | x | x | x | x | x | x | x | x |

x: Undefined

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b3 to b0 | MON1[3:0] | 1 Month | Value for the ones place of months | R/W |
| b4 | MON10 | 10 Months | Value for the tens place of months | R/W |
| b6, b5 | — | Reserved | Set these bits to 0. It is read as the set value. | R/W |
| b7 | ENB | ENB | 0: The register value is not compared with the RMONCNT counter value<br>1: The register value is compared with the RMONCNT counter value | R/W |

RMONAR is an alarm register corresponding to the BCD-coded month counter RMONCNT. When the ENB bit is set to 1, the RMONAR value is compared with the RMONCNT value. From among the alarm registers (RSECAR, RMINAR, RHRAR, RWKAR, RDAYAR, RMONAR, and RYRAREN), only those selected with the ENB bits set to 1 are compared with the corresponding counters. When the respective values all match, the IR flag corresponding to the ALM interrupt is set to 1.
RMONAR values from 01 through 12 (in BCD) can be specified; if a value outside of this range is specified, the RTC does not operate correctly.
This register is cleared to 00h by an RTC software reset.

#### (2) In binary count mode:

Address(es): 0008 C41Ah

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | ENB[15:8] | | | | | | | |
| Value after reset: | x | x | x | x | x | x | x | x |

x: Undefined

The BCNT1AER register is a readable/writable register for setting the alarm enable corresponding to 32-bit binary counter b15 to b8. Among the ENB[31:0] bits, the binary counter (BCNT[31:0]) corresponding to the bits which are set to 1 and the binary alarm register (BCNTAR[31:0]) are compared, and when all match, the IR flag corresponding to the ALM interrupt becomes 1.
This register is cleared to 00h by an RTC software reset.

## 23.2.15 Year Alarm Register (RYRAR)/Binary Counter 2 Alarm Enable Register (BCNT2AER)

### (1) In calendar count mode:

Address(es): 0008 C41Ch

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | YR10[3:0] | | | | YR1[3:0] | | | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | x | x | x | x | x | x | x | x |

x: Undefined

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b3 to b0 | YR1[3:0] | 1 Year | Value for the ones place of years | R/W |
| b7 to b4 | YR10[3:0] | 10 Years | Value for the tens place of years | R/W |
| b15 to b8 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |

RYRAR is an alarm register corresponding to the BCD-coded year counter RYRCNT.
RYRAR values from 00 through 99 (in BCD) can be specified; if a value outside of this range is specified, the RTC does not operate correctly.
This register is cleared to 0000h by an RTC software reset.

### (2) In binary count mode:

Address(es): 0008 C41Ch

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | ENB[23:16] | | | | | | | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | x | x | x | x | x | x | x | x |

x: Undefined

The BCNT2AER register is a readable/writable register for setting the alarm enable corresponding to 32-bit binary counter b23 to b16. Among the ENB[31:0] bits, the binary counter (BCNT[31:0]) corresponding to the bits which are set to 1 and the binary alarm register (BCNTAR[31:0]) are compared, and when all match, the IR flag corresponding to the ALM interrupt becomes 1.
This register is cleared to 0000h by an RTC software reset.

### 23.2.16 Year Alarm Enable Register (RYRAREN)/Binary Counter 3 Alarm Enable Register (BCNT3AER)

(1) In calendar count mode:

Address(es): 0008 C41Eh

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | ENB | — | — | — | — | — | — | — |
| Value after reset: | x | x | x | x | x | x | x | x |

x: Undefined

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b6 to b0 | — | Reserved | Set these bits to 0. It is read as the set value. | R/W |
| b7 | ENB | ENB | 0: The register value is not compared with the RYRCNT counter value<br>1: The register value is compared with the RYRCNT counter value | R/W |

When the ENB bit in RYRAREN is set to 1, the RYRAR value is compared with the RYRCNT value. From among the alarm registers (RSECAR, RMINAR, RHRAR, RWKAR, RDAYAR, RMONAR, and RYRAREN), only those selected with the ENB bits set to 1 are compared with the corresponding counters. When the respective values all match, the IR flag corresponding to the ALM interrupt is set to 1.
This register is cleared to 00h by an RTC software reset.

(2) In binary count mode:

Address(es): 0008 C41Eh

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | ENB[31:24] | | | | | | | |
| Value after reset: | x | x | x | x | x | x | x | x |

x: Undefined

The BCNT3AER register is a readable/writable register for setting the alarm enable corresponding to 32-bit binary counter b31 to b24. Among the ENB[31:0] bits, the binary counter (BCNT[31:0]) corresponding to the bits which are set to 1 and the binary alarm register (BCNTAR[31:0]) are compared, and when all match, the IR flag corresponding to the ALM interrupt becomes 1.
This register is cleared to 00h by an RTC software reset.

## 23.2.17 RTC Control Register 1 (RCR1)

Address(es): 0008 C422h

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | PES[3:0] | | | | RTCOS | PIE | CIE | AIE |
| Value after reset: | x | x | x | x | 0 | x | 0 | x |

x: Undefined

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b0 | AIE | Alarm Interrupt Enable | 0: An alarm interrupt request is disabled.<br>1: An alarm interrupt request is enabled. | R/W |
| b1 | CIE | Carry Interrupt Enable | 0: A carry interrupt request is disabled.<br>1: A carry interrupt request is enabled. | R/W |
| b2 | PIE | Periodic Interrupt Enable | 0: A periodic interrupt request is disabled.<br>1: A periodic interrupt request is enabled. | R/W |
| b3 | RTCOS | RTCOUT Output Select | 0: RTCOUT outputs 1 Hz.<br>1: RTCOUT outputs 64 Hz. | R/W |
| b7 to b4 | PES[3:0] | Periodic Interrupt Select | b7 b4<br>0 1 1 0: A periodic interrupt is generated every 1/256 second.<br>0 1 1 1: A periodic interrupt is generated every 1/128 second.<br>1 0 0 0: A periodic interrupt is generated every 1/64 second.<br>1 0 0 1: A periodic interrupt is generated every 1/32 second.<br>1 0 1 0: A periodic interrupt is generated every 1/16 second.<br>1 0 1 1: A periodic interrupt is generated every 1/8 second.<br>1 1 0 0: A periodic interrupt is generated every 1/4 second.<br>1 1 0 1: A periodic interrupt is generated every 1/2 second.<br>1 1 1 0: A periodic interrupt is generated every 1 second.<br>1 1 1 1: A periodic interrupt is generated every 2 seconds.<br>Other than above: No periodic interrupts are generated. | R/W |

The RCR1 register is used in both calendar count mode and in binary count mode.
Bits AIE, PIE, and PES[3:0] are updated synchronously with the count source. When the RCR1 register is modified, check that all the bits have been updated before proceeding to the next processing.

### AIE Bit (Alarm Interrupt Enable)

This bit enables or disables alarm interrupt requests.

### CIE Bit (Carry Interrupt Enable)

This bit enabled and disables interrupt requests when a carry to the RSECCNT register occurs, or when a carry to the R64CNT register occurs while reading the 64 Hz counter.

### PIE Bit (Periodic Interrupt Enable)

This bit enables or disabled a periodic interrupt.

### RTCOS Bit (RTCOUT Output Select)

This bit is to select the RTCOUT output period. The RTCOS bit must be rewritten while count operation is stopped (the RCR2.START bit is 0) and RTCOUT output is disabled (the RCR2.RTCOE bit is 0). When the RTCOUT is output to an external pin, the RCR2.RTCOE bit must be enabled. For details on controlling I/O ports, refer to section 19.3.1, Procedure for Specifying Input/Output Pin Functions.

### PES[3:0] Bits (Periodic Interrupt Select)

These bits specify the period for the periodic interrupt. A periodic interrupt is generated with the period specified by these bits.

## 23.2.18 RTC Control Register 2 (RCR2)

### (1) In calendar count mode:

Address(es): 0008 C424h

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | CNTMD | HR24 | AADJP | AADJE | RTCOE | ADJ30 | RESET | START |
| Value after reset: | x | x | x | x | 0 | 0 | 0 | x |

x: Undefined

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b0 | START | Start | 0: Prescaler and time counter are stopped.<br>1: Prescaler and time counter operate normally. | R/W |
| b1 | RESET | RTC Software Reset | • In writing<br>0: Writing is invalid.<br>1: The prescaler and the target registers for RTC software reset *1 are initialized<br>• In reading<br>0: In normal time operation, or an RTC software reset has completed.<br>1: During an RTC software reset | R/W |
| b2 | ADJ30 | 30-Second Adjustment | • In writing<br>0: Writing is invalid.<br>1: 30-second adjustment is executed.<br>• In reading<br>0: In normal time operation, or 30-second adjustment has completed.<br>1: During 30-second adjustment | R/W |
| b3 | RTCOE | RTCOUT Output Enable | 0: RTCOUT output disabled.<br>1: RTCOUT output enabled. | R/W |
| b4 | AADJE | Automatic Adjustment Enable | 0: Automatic adjustment is disabled.<br>1: Automatic adjustment is enabled. | R/W |
| b5 | AADJP | Automatic Adjustment Period Select | 0: The RADJ.ADJ[5:0] setting value is adjusted from the count value of the prescaler every minute.<br>1: The RADJ.ADJ[5:0] setting value is adjusted from the count value of the prescaler every 10 seconds. | R/W |
| b6 | HR24 | Hours Mode | 0: The RTC operates in 12-hour mode.<br>1: The RTC operates in 24-hour mode. | R/W |
| b7 | CNTMD | Count Mode Select | 0: The calendar count mode.<br>1: The binary count mode. | R/W |

Note 1. R64CNT, RSECAR/BCNT0AR, RMINAR/BCNT1AR, RHRAR/BCNT2AR, RWKAR/BCNT3AR, RDAYAR/BCNT0AER, RMONAR/BCNT1AER, RYRAR/BCNT2AER, RYRAREN/BCNT3AER, RADJ, RCR2.ADJ30, RCR2.AADJE, RCR2.RTCOE

The RCR2 register is related to hours mode, automatic adjustment function, enabling the RTCOUT output, 30-second adjustment, RTC software reset, and controlling count operation.

**START Bit (Start)**

This bit stops or restarts the prescaler or time counter operation.

The START bit is updated in synchronization with the next count source. When the START bit is modified, check that the bit has been updated before proceeding to the next processing.

**RESET Bit (RTC Software Reset)**

This bit initializes the prescaler and registers to be reset by RTC software.

When 1 is written to the RESET bit, the initialization starts in synchronization with the count source. When the

initialization is completed, the RESET bit is automatically cleared to 0.
When 1 is written to the RESET bit, check that the bit is cleared to 0, and then make next settings.

#### ADJ30 Bit (30-Second Adjustment)

This bit is for 30-second adjustment.
When 1 is written to the ADJ30 bit, the RSECCNT value of 30 seconds or less is rounded down to 00 second and the value of 30 seconds or more is rounded up to 1 minute.
The 30-second adjustment is performed in synchronization with the count source. When 1 is written to this bit, the ADJ30 bit is automatically cleared to 0 after the 30-second adjustment is completed. In case when 1 is written to the ADJ30 bit, check that the bit is cleared to 0, and then make next settings.
When the 30-second adjustment is performed, the prescaler and R64CNT are also reset.
The ADJ30 bit is cleared to 0 by an RTC software reset.

#### RTCOE Bit (RTCOUT Output Enable)

This bit enables output of a 1-Hz/64-Hz clock signal from the RTCOUT pin.
Use the START bit to stop counting by the counters before changing the value of the RTCOE bit. Do not stop counting (write 0 to the START bit) and change the value of the RTCOE bit at the same time.
When RTCOUT is to be output from an external pin, enable the RTCOE bit and set up the port control for the pin.

#### AADJE Bit (Automatic Adjustment Enable)

This bit controls (enables or disables) automatic adjustment.
Set the plus–minus bits (RADJ.PMADJ[1:0]) to 00b (adjustment is not performed) before changing the value of the AADJE bit.
The AADJE bit is cleared to 0 by an RTC software reset.

#### AADJP Bit (Automatic Adjustment Period Select)

This bit selects the automatic-adjustment period.
Set the plus–minus bits (RADJ.PMADJ[1:0]) to 00b (adjustment is not performed) before changing the value of the AADJP bit.
The AADJP bit is cleared to 0 by an RTC software reset.

#### HR24 Bit (Hours Mode)

This bit specifies whether the RTC will operate in 12- or 24-hour mode.
Use the START bit to stop counting by the counters before changing the value of the HR24 bit. Do not stop counting (write 0 to the START bit) and change the value of the HR24 bit at the same time.

#### CNTMD Bit (Count Mode Select)

This bit specifies whether the RTC count mode is operated in calendar count mode or in binary count mode.
When setting the count mode, execute an RTC software reset and start again from the initial settings.
This bit is updated synchronously with the count source, and its value is fixed before the RTC software reset is completed.
For details on initial settings, refer to section 23.3.1, Outline of Initial Settings of Registers after Power On.

(2) In binary count mode:

Address(es): 0008 C424h

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | CNTMD | — | AADJP | AADJE | RTCOE | — | RESET | START |
| Value after reset: | x | x | x | x | 0 | 0 | 0 | x |

x: Undefined

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b0 | START | Start | 0: The 32-bit binary counter, 64-Hz counter, and prescaler are stopped.<br>1: The 32-bit binary counter, 64-Hz counter, and prescaler are in normal operation. | R/W |
| b1 | RESET | RTC Software Reset | • In writing<br>0: Writing is invalid.<br>1: The prescaler and the target registers for RTC software reset *1 are initialized<br>• In reading<br>0: In normal time operation, or an RTC software reset has completed.<br>1: During an RTC software reset | R/W |
| b2 | — | Reserved | This bit is read as 0. The write value should be 0. | R/W |
| b3 | RTCOE | RTCOUT Output Enable | 0: RTCOUT output disabled.<br>1: RTCOUT output enabled. | R/W |
| b4 | AADJE | Automatic Adjustment Enable | 0: Automatic adjustment is disabled.<br>1: Automatic adjustment is enabled. | R/W |
| b5 | AADJP | Automatic Adjustment Period Select | 0: Adds or subtracts the RADJ.ADJ[5:0] bits from the prescaler count value every 32 seconds<br>1: Adds or subtracts the RADJ.ADJ[5:0] bits from the prescaler count value every 8 seconds | R/W |
| b6 | — | Reserved | This bit is undefined. The write value should be 0. | R/W |
| b7 | CNTMD | Count Mode Select | 0: The calendar count mode.<br>1: The binary count mode. | R/W |

Note 1. R64CNT, RSECAR/BCNT0AR, RMINAR/BCNT1AR, RHRAR/BCNT2AR, RWKAR/BCNT3AR, RDAYAR/BCNT0AER, RMONAR/BCNT1AER, RYRAR/BCNT2AER, RYRAREN/BCNT3AER, RADJ, RCR2.ADJ30, RCR2.AADJE, RCR2.RTCOE

### START Bit (Start)

This bit stops or restarts the prescaler or counter (clock) operation.

The START bit is updated in synchronization with the count source. When the START bit is modified, check that the bit is updated without fail, and then make next settings.

### RESET Bit (RTC Software Reset)

This bit initializes the prescaler and registers to be reset by RTC software.

When 1 is written to the RESET bit, the initialization starts in synchronization with the count source. When the initialization is completed, the RESET bit is automatically cleared to 0.

When 1 is written to the RESET bit, check that the bit is cleared to 0, and then make next settings.

#### RTCOE Bit (RTCOUT Output Enable)

This bit enables output of a 1-Hz/64-Hz clock signal from the RTCOUT pin.
Use the START bit to stop counting by the counters before changing the value of the RTCOE bit. Do not stop counting (write 0 to the START bit) and change the value of the RTCOE bit at the same time.
When an RTCOUT signal is to be output from an external pin, enable the port control as well as setting this bit.

#### AADJE Bit (Automatic Adjustment Enable)

This bit controls (enables or disables) automatic adjustment.
Set the plus–minus bits (RADJ.PMADJ[1:0]) to 00b (adjustment is not performed) before changing the value of the AADJE bit.
The AADJE bit is cleared to 0 by an RTC software reset.

#### AADJP Bit (Automatic Adjustment Period Select)

This bit selects the automatic-adjustment period.
Correction period can be selected from 32 second units or 8 second units in binary count mode.
Set the plus–minus bits (RADJ.PMADJ[1:0]) to 00b (adjustment is not performed) before changing the value of the AADJP bit.
The AADJP bit is cleared to 0 by an RTC software reset.

#### CNTMD Bit (Count Mode Select)

This bit specifies whether the RTC count mode is operated in calendar count mode or in binary count mode.
When setting the count mode, execute an RTC software reset and start again from the initial settings.
This bit is updated synchronously with the count source, and its value is fixed before the RTC software reset is completed.
For details on initial settings, refer to section 23.3.1, Outline of Initial Settings of Registers after Power On.

## 23.2.19 RTC Control Register 3 (RCR3)

Address(es): 0008 C426h

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | RTCDV[2:0] | | | RTCEN |
| Value after reset: | 0 | 0 | 0 | 0 | x | x | x | x |

x: Undefined

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b0 | RTCEN | Sub-clock Oscillator Control | 0: Sub-clock oscillator is stopped.<br>1: Sub-clock oscillator is operating. | R/W |
| b3 to b1 | RTCDV[2:0] | Sub-clock Oscillator Drive Capacity Control | b3 b1<br>0 0 0: Medium drive capacity (4.4 pF type)<br>0 0 1: High drive capacity (6.0 pF type)<br>1 0 1: Low drive capacity (3.7 pF type)<br>Settings other than above are prohibited. | R/W |
| b7 to b4 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |

The RCR3 register is used for controlling the sub-clock oscillator in the clock generation circuit. For details on controlling the sub-clock oscillator, refer to section 9, Clock Generation Circuit.
This register is a function common to calendar count mode and binary count mode.
When this register is modified, check that all the bits have been updated before proceeding to the next processing.

### RTCEN Bit (Sub-clock Oscillator Control)

The RTCEN bit and a clock generation circuit register control whether to operate or stop the sub-clock oscillator. If one of the bits is set so as to enable the operation, the sub-clock oscillator runs.
When using the sub-clock as the count source to the RTC, set up the sub-clock oscillator using the RTCEN bit.

### RTCDV[2:0] Bits (Sub-Clock Oscillator Drive Capacity Control)

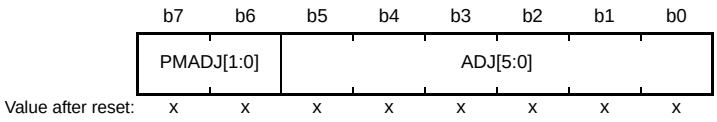
These bits control the drive capacity of the sub-clock oscillator. Set the RTCDV[2:0] bits when the SOSCCR.SOSTP bit is 1 and the RCR3.RTCEN bit is 0.

#### (1) Notes on using a low CL crystal unit

When the signal level of any pin near the XCIN or XCOUT pin is changed, the oscillation accuracy of the sub-clock oscillator may be affected. The accuracy is affected differently depending on the board traces and how the signal level of any pin near the XCIN or XCOUT pin is changed. When designing a board using a low CL crystal unit, refer to the application note “Design Guide for Low CL Sub-clock Circuits” (R01AN1012EJ0100) to reduce the influence from noise.

## 23.2.20 Time Error Adjustment Register (RADJ)

Address(es): 0008 C42Eh




x: Undefined

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b5 to b0 | ADJ[5:0] | Adjustment Value | These bits specify the adjustment value from the prescaler. | R/W |
| b7, b6 | PMADJ[1:0] | Plus–Minus | b7 b6<br>0 0: Adjustment is not performed.<br>0 1: Adjustment is performed by the addition to the prescaler.<br>1 0: Adjustment is performed by the subtraction from the prescaler.<br>1 1: Setting prohibited | R/W |

Adjustment is performed by the addition to or subtraction from the prescaler.
In case when the automatic adjustment enable (RCR2.AADJE) bit is 0, adjustment is performed when writing to the RADJ.
In case when the RCR2.AADJE bit is 1, adjustment is performed in the interval specified by the automatic adjustment period select (RCR2.AADJP) bit.
The current adjustment by software (disabling automatic adjustment) may be invalid if the following adjustment value is specified within 320 cycles of the count source after the register setting. To perform adjustment consecutively, wait for 320 cycles or more of the count source after the register setting and then specify the next adjustment value.
RADJ is updated in synchronization with the count source. When RADJ is modified, check that all the bits have been updated without fail before continuing with further processing.
This register is cleared to 00h by an RTC software reset.

### ADJ[5:0] Bits (Adjustment Value)

These bits specify the adjustment value (the number of sub-clock cycles) from the prescaler.

### PMADJ[1:0] Bits (Plus–Minus)

These bits select whether the clock is set ahead or back depending on the error-adjustment value set in the ADJ[5:0] bits.

## 23.3 Operation

### 23.3.1 Outline of Initial Settings of Registers after Power On

After the power is turned on, the initial settings for the clock setting, time error adjustment, time setting, alarm, interrupt, and time capture control register should be performed.

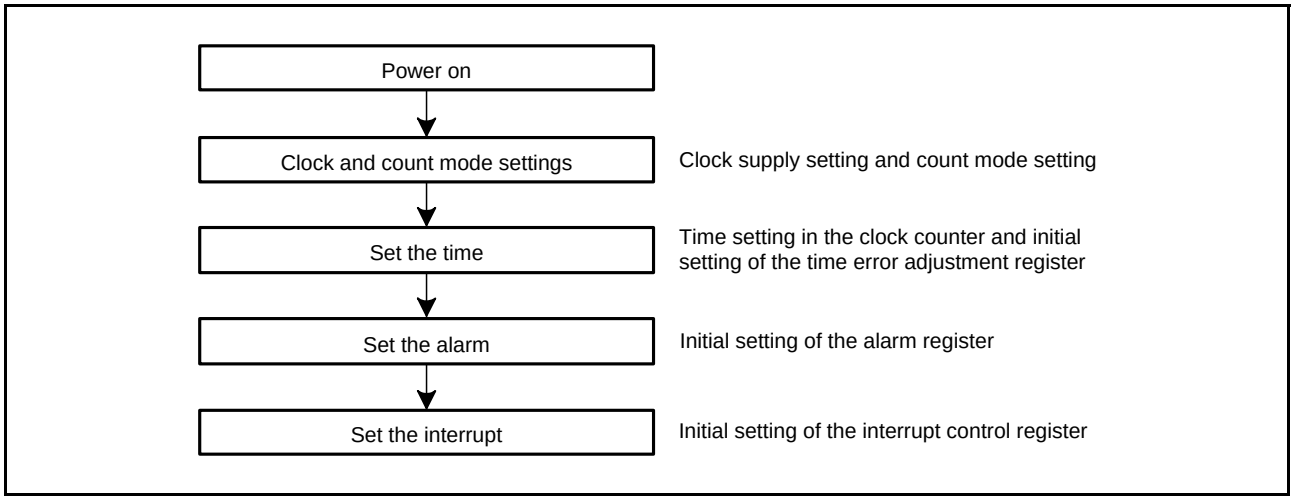
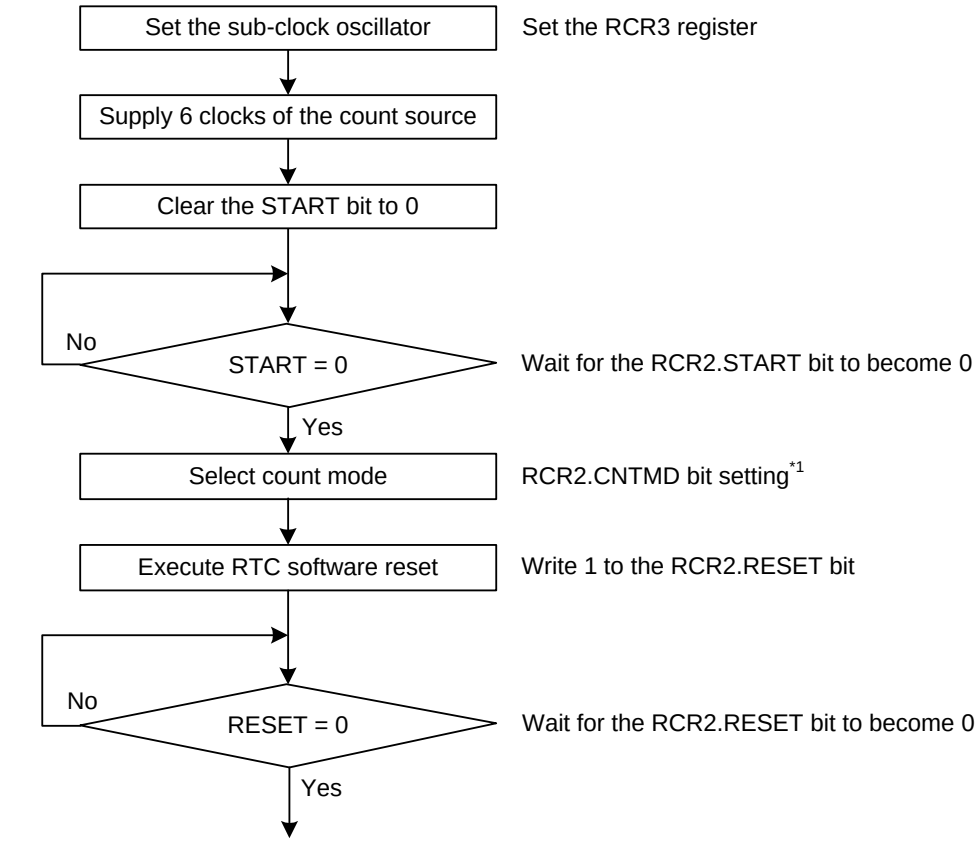



**Figure 23.2 Outline of Initial Settings after Power On**

### 23.3.2 Clock and Count Mode Setting Procedure

Figure 23.3 shows how to set the clock and the count mode.




Note 1. This step is not necessary if the count mode has been set concurrently with setting the START bit to 0. A value corresponding to the count mode setting must be written to the RCR2.CNTMD bit.

**Figure 23.3 Clock and Count Mode Setting Procedure**

### 23.3.3 Setting the Time

Figure 23.4 shows how to set the time.

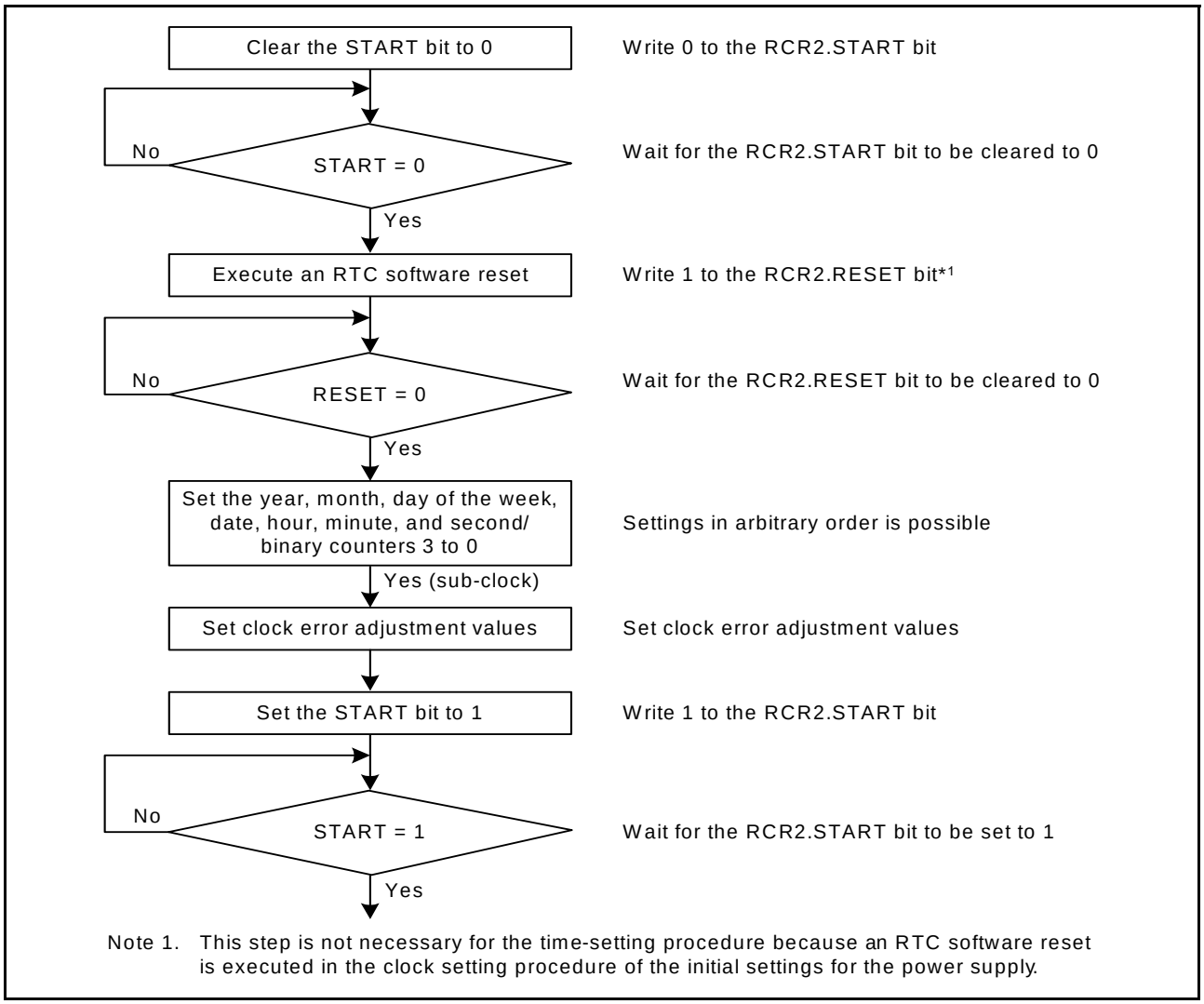



**Figure 23.4 Setting the Time**

### 23.3.4 30-Second Adjustment

Figure 23.5 shows how to execute 30-second adjustment.

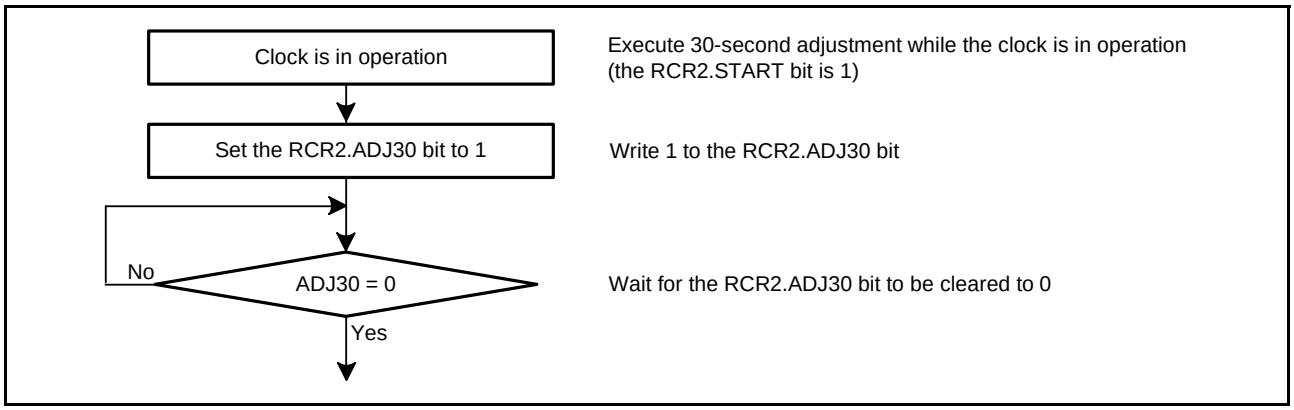



**Figure 23.5 30-Second Adjustment**

### 23.3.5 Reading 64-Hz Counter and Time

Figure 23.6 shows how to read the 64-Hz counter and time.

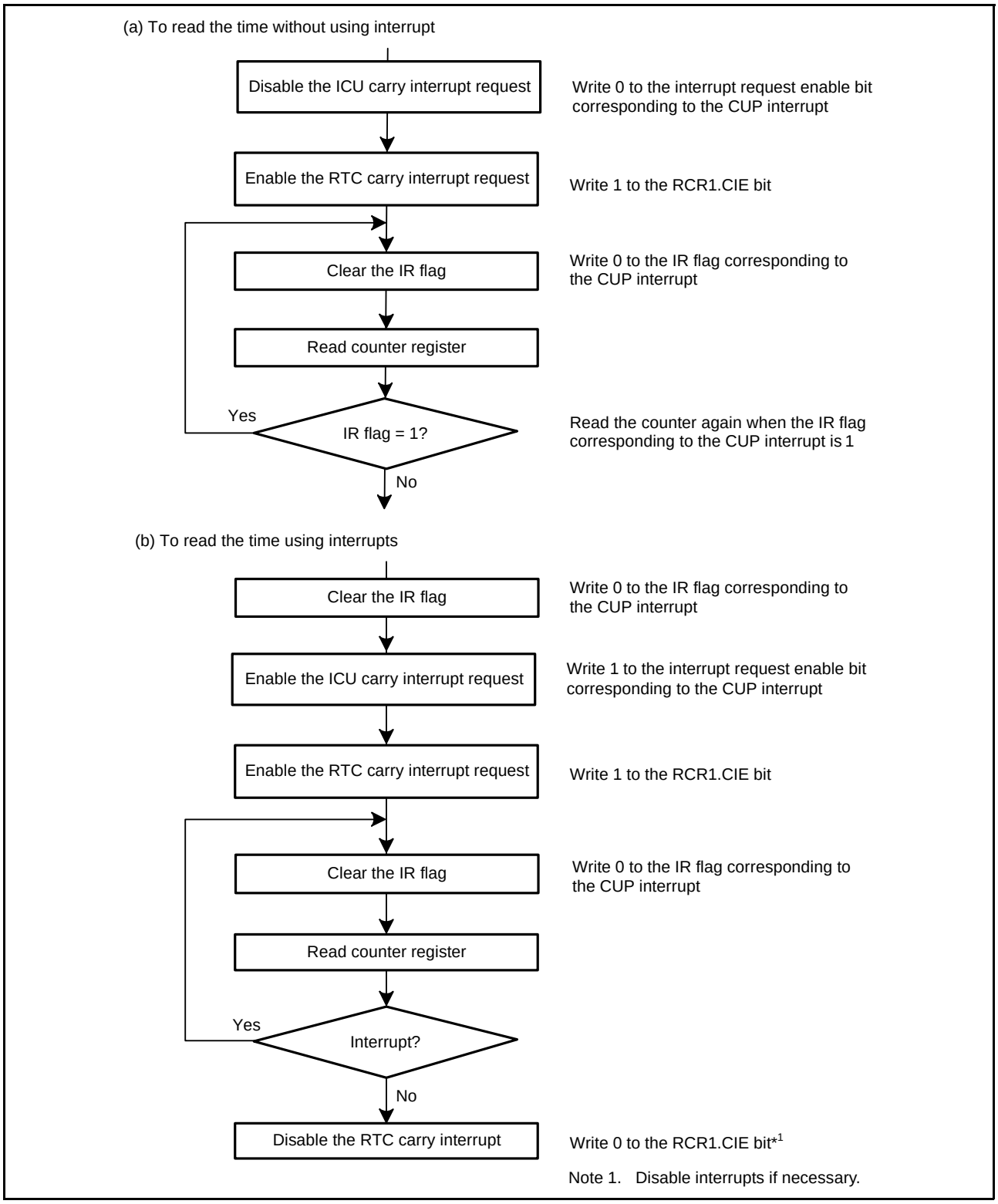



**Figure 23.6 Reading Time**

If a carry occurs while the 64-Hz counter and time are being read, the correct time will not be obtained, so they must be read again. The procedure for reading the time without using interrupts is shown in (a) in Figure 23.6, and the procedure using carry interrupts in (b). To keep the program simple, method (a) should be used in most cases.

### 23.3.6 Alarm Function

Figure 23.7 shows how to use the alarm function.

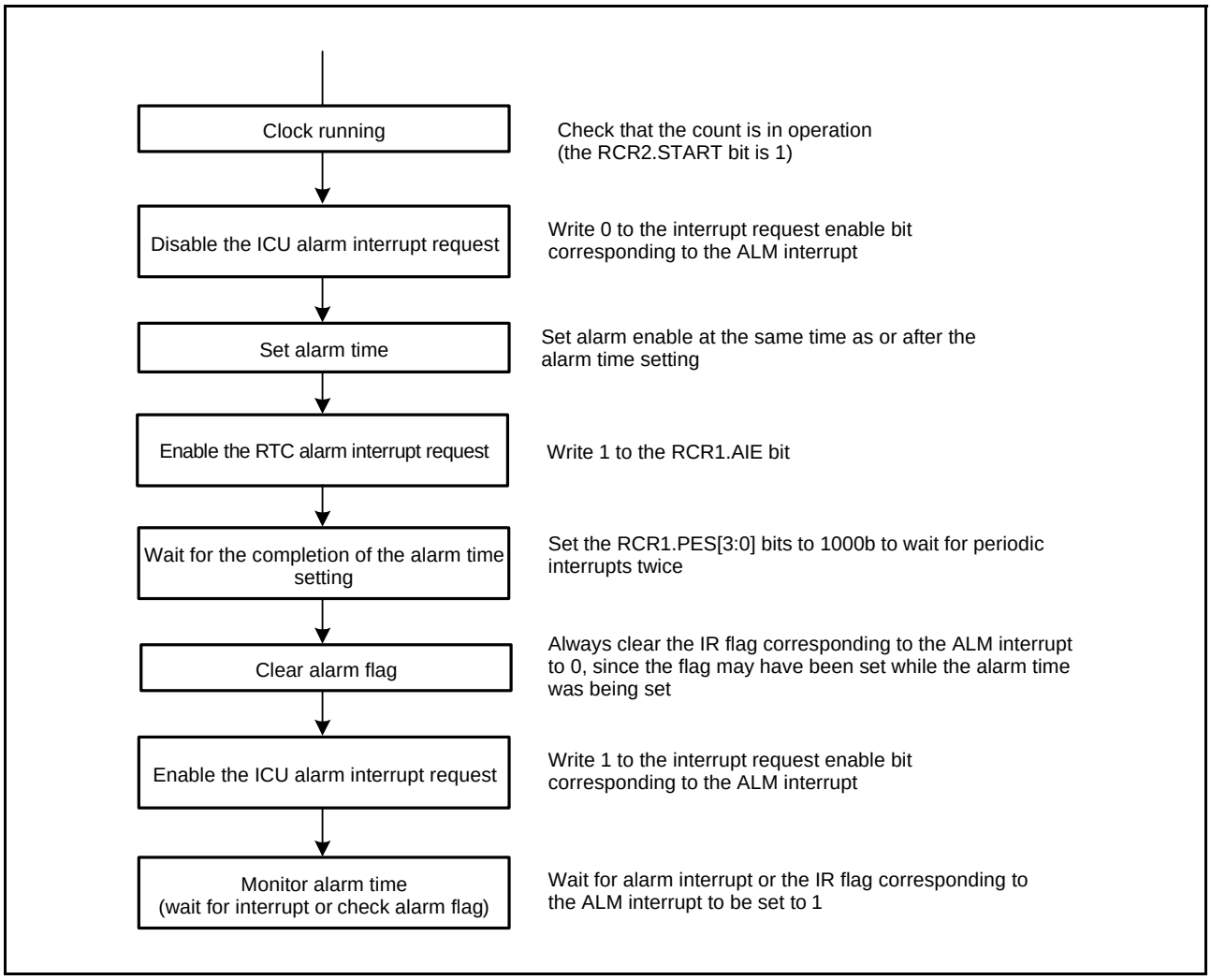



**Figure 23.7 Using Alarm Function**

In calendar count mode, an alarm can be generated by any one of year, month, date, day-of-week, hour, minute or second, or any combination of those. Write 1 to the ENB bit in the alarm registers involved in the alarm setting, and set the alarm time in the lower bits. Write 0 to the ENB bit in registers not involved in the alarm setting.

In binary count mode, an alarm can be generated in any bit combination of 32 bits. Write 1 to the ENB bit of the alarm enable register corresponding to the target bit of the alarm, and set the alarm time to the alarm register. For bits that are not target of the alarm, write 0 to the ENB bit of the alarm enable register.

When the counter and the alarm time match, the IR flag corresponding to the ALM interrupt is set to 1. Alarm detection can be confirmed by reading this bit, but an interrupt should be used in most cases. If 1 has been set in the interrupt request enable bit corresponding to the ALM interrupt, an alarm interrupt is generated in the event of alarm, enabling the alarm to be detected.

Writing 0 clears the IR flag corresponding to the ALM interrupt.

When the counter and the alarm time match in a low power consumption state, the MCU returns from the low power consumption state.

### 23.3.7 Procedure for Disabling Alarm Interrupt

Figure 23.8 shows the procedure for disabling the enabled alarm interrupt request.

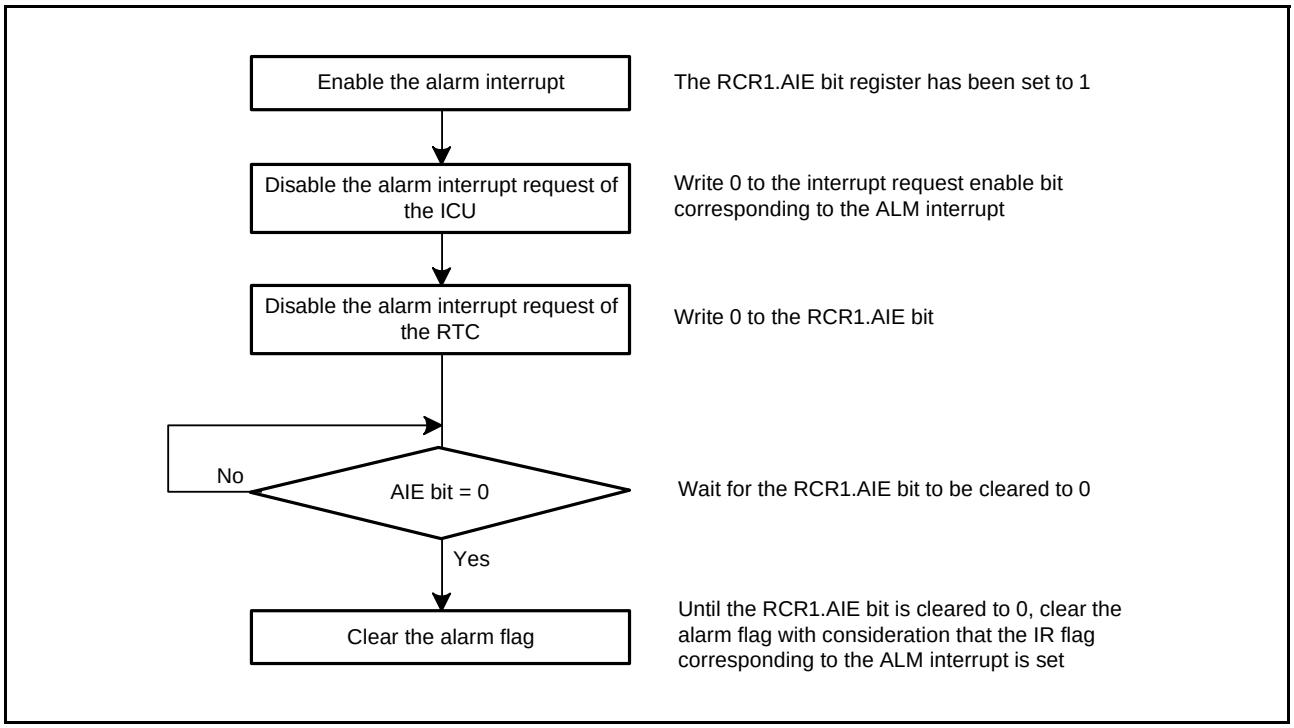



**Figure 23.8 Procedure for Disabling Alarm Interrupt Request**

### 23.3.8 Time Error Adjustment Function

The time error adjustment function is used to correct errors (running fast or slow) in the time due to the precision of oscillation by the sub-clock. Since 32,768 cycles of the sub-clock constitute 1 second of operation when the sub-clock is selected, the clock runs fast if the sub-clock frequency is high and slow if the sub-clock frequency is low. This function can be used to correct errors due to the clock running fast or slow.
Two types of time error adjustment functions are provided: automatic adjustment and adjustment by software.
Use the RCR2.AADJE bit to select automatic adjustment or adjustment by software.

#### 23.3.8.1 Automatic Adjustment

Enable automatic adjustment by setting the RCR2.AADJE bit to 1.
Automatic adjustment is the addition or subtraction of the value counted by the prescaler to or from the value in the RADJ register every time the adjustment period selected by the RCR2.AADJE bit elapses.
Examples are shown below.

[Example 1] Sub-clock running at 32.769 kHz
Adjustment procedure:
When the sub-clock is running at 32.769 kHz, 1 second elapses every 32,769 clock cycles. The RTC is meant to run at 32,768 clock cycles, so the clock runs fast by one clock cycle every second. The time on the clock is fast by 60 clock cycles per minute, so adjustment can take the form of setting the clock back by 60 cycles every minute.

Register settings: (when RCR2.CNTMD = 0)

- RCR2.AADJP = 0 (adjustment every minute)
- RADJ.PMADJ[1:0] = 10b (adjustment is performed by the subtraction from the prescaler.)
- RADJ.ADJ[5:0] = 60 (3Ch)

[Example 2] Sub-clock running at 32.766 kHz
Adjustment procedure:
When the sub-clock is running at 32.766 kHz, 1 second elapses every 32,766 clock cycles. The RTC is meant to run at 32,768 clock cycles, so the clock runs slow by two clock cycles every second. The time on the clock is slow by 20 clock cycles every 10 seconds, so adjustment can take the form of setting the clock forward by 20 cycles every 10 seconds.

Register settings: (when RCR2.CNTMD = 0)

- RCR2.AADJP = 1 (adjustment every 10 seconds)
- RADJ.PMADJ[1:0] = 01b (adjustment is performed by the addition to the prescaler.)
- RADJ.ADJ[5:0] = 20 (14h)

[Example 3] Sub-clock running at 32.764 kHz
Adjustment procedure:
At 32.764 kHz, 1 second elapses on 32,764 clock cycles. Since the RTC operates for 32,768 clock cycles as 1 second, the clock is delayed for four clock cycles per second. In 8 seconds, the delay is 32 clock cycles, so correction can be made by proceeding the clock for 32 clock cycles every 8 seconds.

Register settings when the RCR2.CNTMD bit is 1

- RCR2.AADJP = 1 (adjustment every 8 seconds)
- RADJ.PMADJ[1:0] = 01b (adjustment is performed by the addition to the prescaler.)
- RADJ.ADJ[5:0] = 32 (20h)

#### 23.3.8.2 Adjustment by Software

Enable adjustment by software by setting the RCR2.AADJE bit to 0.
Adjustment by software is the addition or subtraction of the value counted by the prescaler to or from the value in the RADJ register at the time of execution of an instruction for writing to the RADJ register.
An example is shown below.

[Example 1] Sub-clock running at 32.769 kHz
Adjustment procedure:
When the sub-clock is running at 32.769 kHz, 1 second elapses every 32,769 clock cycles. The RTC is meant to run at 32,768 clock cycles, so the clock runs fast by one clock cycle every second. The time on the clock is fast by one clock cycle per second, so adjustment can take the form of setting the clock back by one cycle every second.

Register settings:

- RADJ.PMADJ[1:0] = 10b (adjustment is performed by the subtraction from the prescaler.)
- RADJ.ADJ[5:0] = 1 (01h)
  This is written to the RADJ register once per 1-second interrupt.

#### 23.3.8.3 Procedure for Changing the Mode of Adjustment

When changing the mode of adjustment, change the value of the AADJE bit in RCR2 after clearing the RADJ.PMADJ[1:0] bits to 00b (adjustment is not performed).

Changing from adjustment by software to automatic adjustment:

(1) Clear the RADJ.PMADJ[1:0] bits to 00b (adjustment is not performed).
(2) Set the RCR2.AADJE bit to 1 (enabling automatic adjustment).
(3) Use the RCR2.AADJP bit to select the period of adjustment.
(4) In RADJ, set the PMADJ[1:0] bits for addition or subtraction and the ADJ[5:0] bits to the value for use in time error adjustment.

Changing from adjustment by software to automatic adjustment:

(1) Clear the RADJ.PMADJ[1:0] bits to 00b (adjustment is not performed).
(2) Set the RCR2.AADJE bit to 0 (enabling adjustment by software).
(3) Proceed with adjustment by setting the RADJ.PMADJ[1:0] bits for addition or subtraction and the RADJ.ADJ[5:0] bits to the value for use in time error adjustment at the desired time. After that, the time is adjusted every time a value is written to the RADJ register.

#### 23.3.8.4 Procedure for Stopping Adjustment

Stop adjustment by clearing the RADJ.PMADJ[1:0] bits to 00b (adjustment is not performed).

## 23.4 Interrupt Sources

There are three interrupt sources in the realtime clock. Table 23.3 lists interrupt sources for the RTC.

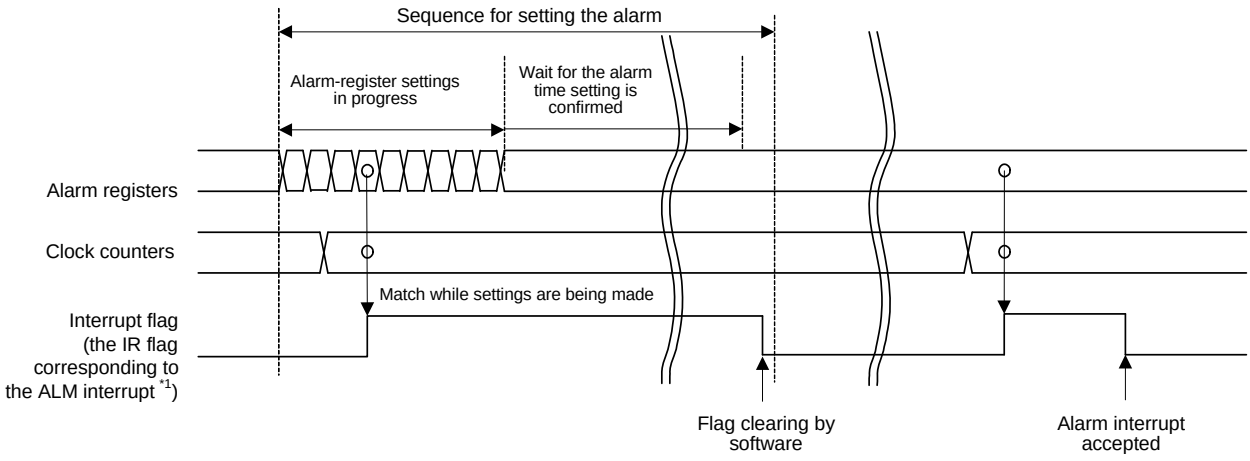
**Table 23.3 RTC Interrupt Sources**

| Name | Interrupt Sources |
|---|---|
| ALM | Alarm interrupt |
| PRD | Periodic interrupt |
| CUP | Carry interrupt |

### (1) Alarm interrupt (ALM)

This interrupt is generated according to the result of comparison between the alarm registers and realtime clock counters (for details, refer to section 23.3.6, Alarm Function).

Since there is a possibility of the interrupt flag being set when the settings of the alarm registers match the clock counters, wait for the alarm time settings to be confirmed and clear the IR flag corresponding to the ALM interrupt to 0 again after modifying values of the alarm registers. Once the interrupt flag for the alarm interrupt has been cleared and the state has returned to non-matching of the alarm registers and clock counters, the flag will not be set again until there is a further match or the values of the alarm registers are modified again.




Note 1. Refer to the Interrupt Controller section for details on the corresponding interrupt vector number.

**Figure 23.9 Timing Chart for the Alarm Interrupt (ALM)**

### (2) Periodic interrupt (PRD)

This interrupt is generated at intervals of 2 seconds, 1 second, 1/2 second, 1/4 second, 1/8 second, 1/16 second, 1/32 second, 1/64 second, 1/128 second, or 1/256 second. The interrupt interval can be selected through the RCR1.PES[3:0] bits.

### (3) Carry interrupt (CUP)

This interrupt is generated when a carry to the second counter/binary counter 0 occurred or a carry to the R64CNT counter occurred during read access to the 64-Hz counter.

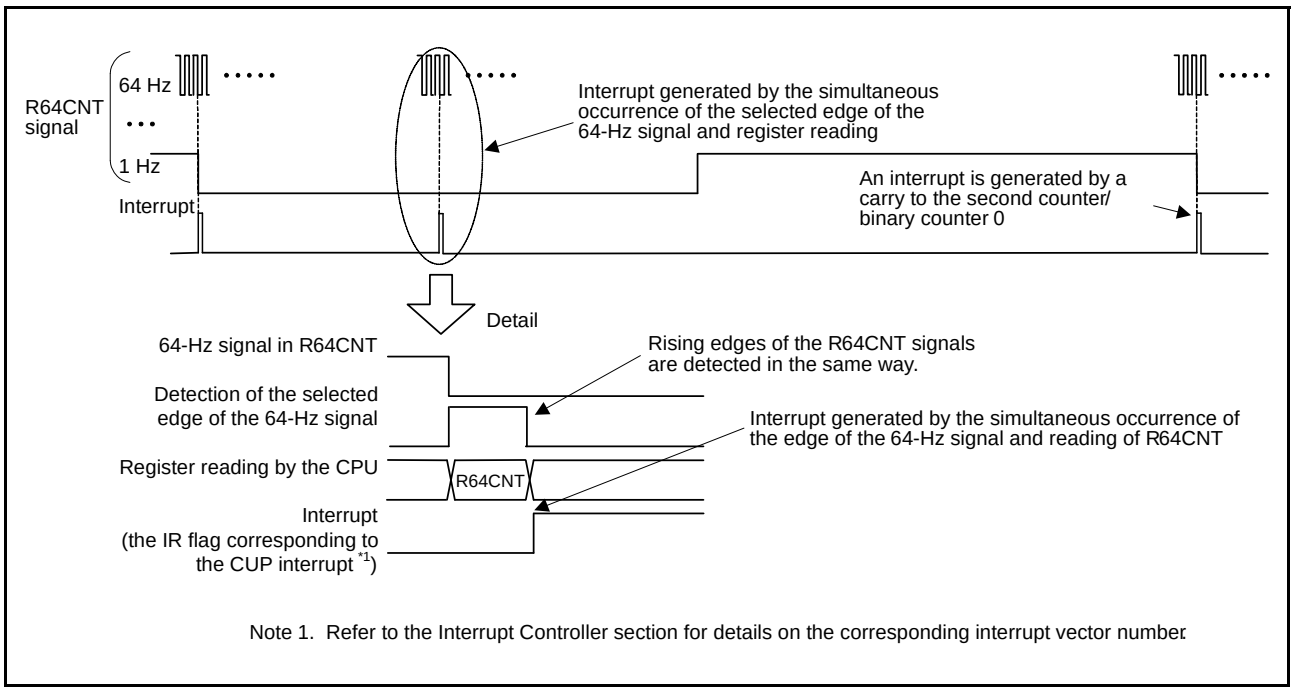



**Figure 23.10 Carry Interrupt (CUP) Timing Chart**

## 23.5 Usage Notes

### 23.5.1 Register Writing during Counting

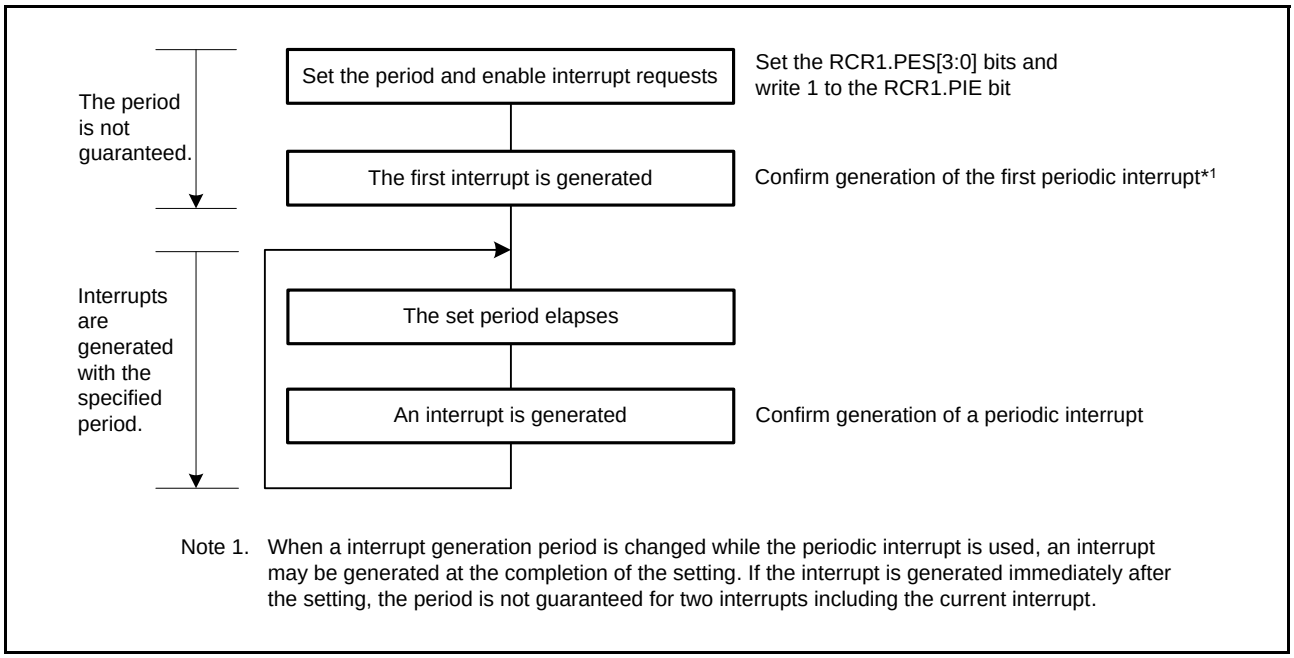
The following registers should not be written to during counting (while the RCR2.START bit = 1).

RSECCNT/BCNT0, RMINCNT/BCNT1, RHRCNT/BCNT2, RDAYCNT, RWKCNT/BCNT3, RMONCNT, RYRCNT, RCR1.RTCOS, RCR2.RTCOE, RCR2.HR24

The counter must be stopped before writing to any of the above registers.

### 23.5.2 Use of Periodic Interrupts

The procedure for using periodic interrupts is shown in Figure 23.11.
The generation and period of the periodic interrupt can be changed by the setting of the RCR1.PES[3:0] bits. However, since the prescaler, R64CNT, and RSECCNT/BCNT0 are used to generate interrupts, the interrupt period is not guaranteed immediately after setting of the RCR1.PES[3:0] bits.
Furthermore, stopping/restarting or resetting counter operation, reset by RTC software, and the 30-second adjustment by changing the RCR2 value affects the interrupt period. When the time error adjustment function is used, the interrupt generation period after adjustment is added or subtracted according to the adjustment value.




Note 1. When a interrupt generation period is changed while the periodic interrupt is used, an interrupt may be generated at the completion of the setting. If the interrupt is generated immediately after the setting, the period is not guaranteed for two interrupts including the current interrupt.

**Figure 23.11 Using Periodic Interrupt Function**

### 23.5.3 RTCOUT (1-Hz/64-Hz) Clock Output

Stopping/restarting or resetting counter operation, reset by RTC software, and the 30-second adjustment by changing the RCR2 value affects the period of RTCOUT (1-Hz/64-Hz) output. When the time error adjustment function is used, the period of RTCOUT (1-Hz/64-Hz) output after adjustment is added or subtracted according to the adjustment value.

### 23.5.4 Transitions to Low Power Consumption Modes after Setting Registers

A transition to a low power consumption state (software standby mode) during writing to or updating of an RTC register might destroy the register's value. After setting a register, confirm that the setting is in place before initiating a transition to a low power consumption state.

### 23.5.5 Points for Caution when Writing to and Reading from Registers

- When reading a counter register such as the second counter/binary counter after having written to the counter register, follow the procedure in section 23.3.5, Reading 64-Hz Counter and Time.
- The value written to the count registers, alarm registers, year alarm enable register, bits RCR2.AADJE, AADJP, and HR24, or RCR3 register is reflected when four read operations are performed after writing.
- The values written to the RCR1.CIE, RTCOS, and RCR2.RTCOE bits can be read immediately after writing.
- To read the value from the timer counter after return from a reset, or period in software standby mode, wait for 1/128 second while the clock is operating (RCR2.START bit = 1).
- After a reset is generated, write to the RTC register when six cycles of the count source clock have elapsed.

### 23.5.6 Changing the Count Mode

When changing the count mode (calendar/binary), set the RCR2.START bit to 0, stop counting operation, then start again from the initial setting. For details on initial setting, refer to section 23.3.1, Outline of Initial Settings of Registers after Power On.

### 23.5.7 Initialization Procedure when the Realtime Clock is not to be Used

Registers in the RTC are not initialized by a reset. Accordingly, depending on the initial state, the generation of an unintentional interrupt or operation of the counter may lead to increased power consumption.
For products that do not require a realtime clock, initialize the registers by following the initialization procedure shown in Figure 23.12.

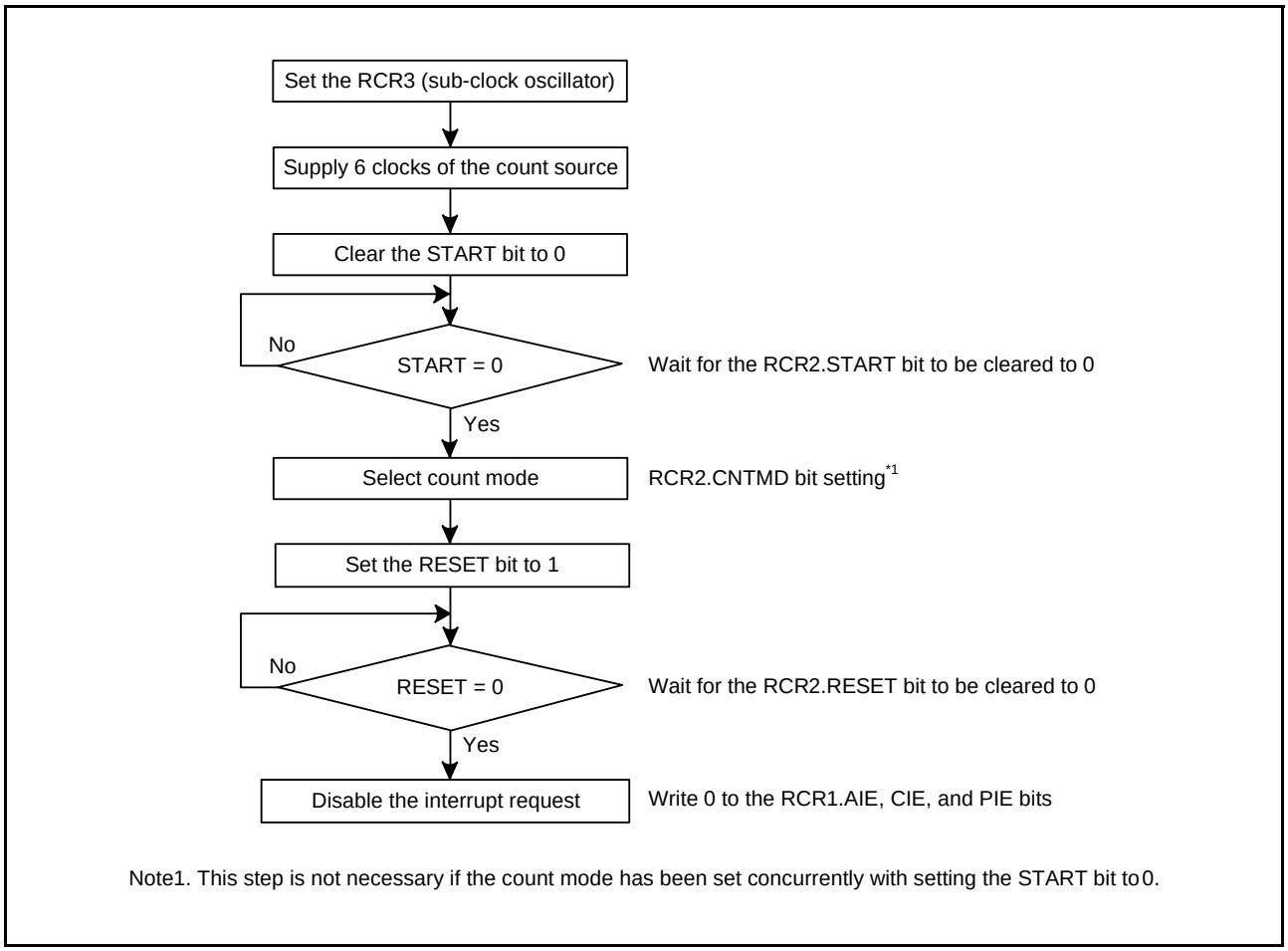



**Figure 23.12 Initialization Procedure**

# 24. Independent Watchdog Timer (IWDTa)

## 24.1 Overview

The independent watchdog timer (IWDT) can be used to detect programs being out of control.
The user can detect when a program runs out of control if an underflow occurs, by creating a program that refreshes the IWDT counter before it underflows.
The functions of the IWDT are different from those of the WDT in the following respects.

- The divided IWDT-dedicated low-speed clock (IWDTCLK) is used as the count source (not affected by the PCLK)
- When making a transition to sleep mode, software standby mode, or deep sleep mode, the IWDTCSTPR.SLCSTP bit can be used to select whether to stop the counter or not.

Table 24.1 lists the specifications of the IWDT and Figure 24.1 shows a block diagram of the IWDT.

**Table 24.1 IWDT Specifications**

| Item | Description |
|---|---|
| Count source*1 | IWDT-dedicated clock (IWDTCLK) |
| Clock division ratio | Division by 1, 16, 32, 64, 128, or 256 |
| Counter operation | Counting down using a 14-bit down-counter |
| Conditions for starting the counter | • Counting automatically starts after a reset (auto-start mode)<br>• Counting is started (register start mode) by refreshing the counter (writing 00h and then FFh to the IWDTRR register). |
| Conditions for stopping the counter | • Reset (the down-counter and other registers return to their initial values)<br>• A counter underflows or a refresh error is generated<br>Count restarts automatically in auto-start mode, or by refreshing the counter in register start mode |
| Window function | Window start and end positions can be specified (refresh-permitted and refresh-prohibited periods) |
| Reset-output sources | • Down-counter underflows<br>• Refreshing outside the refresh-permitted period (refresh error) |
| Interrupt request output sources | • A non-maskable interrupt (WUNI) is generated by an underflow of the down-counter<br>• When refreshing is done outside the refresh-permitted period (refresh error) |
| Reading the counter value | The down-counter value can be read by the IWDTSR register. |
| Output signal (internal signal) | • Reset output<br>• Interrupt request output<br>• Sleep-mode count stop control output |
| Auto-start mode (controlled by option function select register 0 (OFS0)) | • Selecting the clock frequency division ratio after a reset (OFS0.IWDTCKS[3:0] bits)<br>• Selecting the timeout period of the watchdog timer (OFS0.IWDTTOPS[1:0] bits)<br>• Selecting the window start position in the watchdog timer (OFS0.IWDTRPSS[1:0] bits)<br>• Selecting the window end position in the watchdog timer (OFS0.IWDTRPES[1:0] bits)<br>• Selecting the reset output or interrupt request output (OFS0.IWDTRSTIRQS bit)<br>• Selecting the down-count stop function at transition to sleep mode, software standby mode, or deep sleep mode (OFS0.IWDTSLCSTP bit) |
| Register start mode (controlled by the IWDT registers) | • Selecting the clock frequency division ratio after refreshing (IWDTCR.CKS[3:0] bits)<br>• Selecting the timeout period of the watchdog timer (IWDTCR.TOPS[1:0] bits)<br>• Selecting the window start position in the watchdog timer (IWDTCR.RPSS[1:0] bits)<br>• Selecting the window end position in the watchdog timer (IWDTCR.RPES[1:0] bits)<br>• Selecting the reset output or interrupt request output (IWDTRCR.RSTIRQS bit)<br>• Selecting the down-count stop function at transition to sleep mode, software standby mode, or deep sleep mode (IWDTCSTPR.SLCSTP bit) |

Note 1. Satisfy the frequency of the peripheral module clock (PCLKB) ≥ 4 × (the frequency of the count clock source after division).

To use the IWDT, the IWDT-dedicated clock (IWDTCLK) should be supplied so that the IWDT operates even if the peripheral clock (PCLK) stops. The bus interface and registers operate with PCLK, and the 14-bit down-counter and control circuits operate with IWDTCLK.
Figure 24.1 is a block diagram of the IWDT.

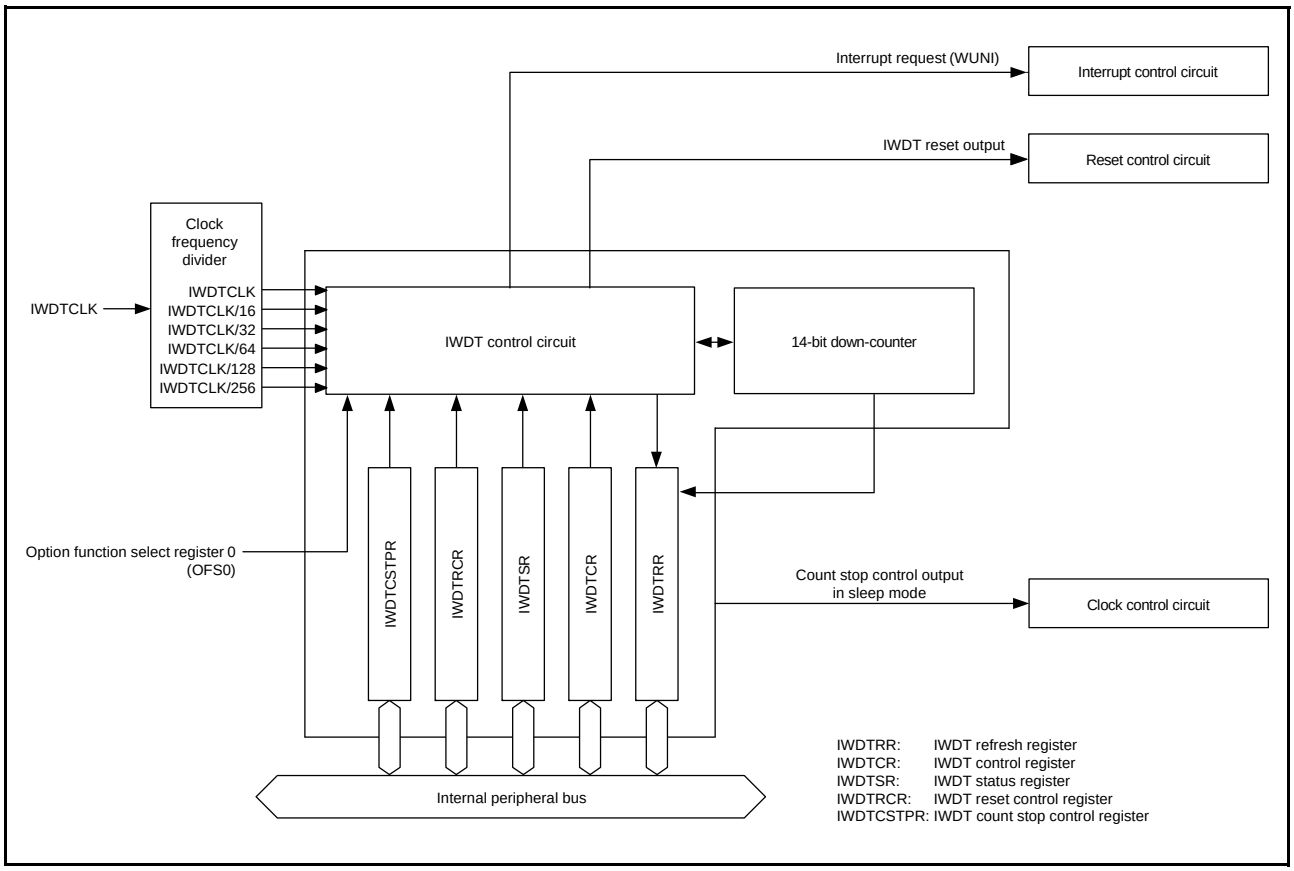
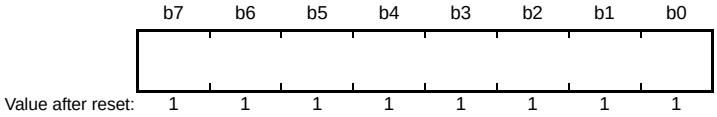



**Figure 24.1 IWDT Block Diagram**

## 24.2 Register Descriptions

### 24.2.1 IWDT Refresh Register (IWDTRR)

Address(es): 0008 8030h




| Bit | Description | R/W |
|---|---|---|
| b7 to b0 | The down-counter is refreshed by writing 00h and then writing FFh to this register. | R/W |

IWDTRR refreshes the down-counter of the IWDT.
The down-counter of the IWDT is refreshed by writing 00h and then writing FFh to IWDTRR (refresh operation) within the refresh-permitted period.
After the down-counter has been refreshed, it starts counting down from the value selected by the IWDT timeout period select bits (OFS0.IWDTTOPS[1:0]) in option function select register 0 (OFS0) in auto-start mode. In register start mode, counting down starts from the value selected by setting the timeout period select (TOPS[1:0]) bits in the IWDT control register (IWDTCR) in the first refresh operation after release from the reset state.
When 00h is written, the read value is 00h. When a value other than 00h is written, the read value FFh.
For details of the refresh operation, refer to section 24.3.3, Refresh Operation.

## 24.2.2 IWDT Control Register (IWDTCR)

Address(es): 0008 8032h

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | RPSS[1:0] | | — | — | RPES[1:0] | | CKS[3:0] | | | | — | — | TOPS[1:0] | |
| Value after reset: | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 1 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b1, b0 | TOPS[1:0] | Timeout Period Selection | b1 b0<br>0 0: 128 cycles (007Fh)<br>0 1: 512 cycles (01FFh)<br>1 0: 1024 cycles (03FFh)<br>1 1: 2048 cycles (07FFh) | R/W |
| b3, b2 | — | Reserved | These bits are read as 0. Writing to these bits has no effect. | R |
| b7 to b4 | CKS[3:0] | Clock Division Ratio Selection | b7 b4<br>0 0 0 0: IWDTCLK<br>0 0 1 0: IWDTCLK/16<br>0 0 1 1: IWDTCLK/32<br>0 1 0 0: IWDTCLK/64<br>1 1 1 1: IWDTCLK/128<br>0 1 0 1: IWDTCLK/256<br>Other settings are prohibited. | R/W |
| b9, b8 | RPES[1:0] | Window End Position Selection | b9 b8<br>0 0: 75%<br>0 1: 50%<br>1 0: 25%<br>1 1: 0% (window end position is not specified.) | R/W |
| b11, b10 | — | Reserved | These bits are read as 0. Writing to these bits has no effect. | R |
| b13, b12 | RPSS[1:0] | Window Start Position Selection | b13 b12<br>0 0: 25%<br>0 1: 50%<br>1 0: 75%<br>1 1: 100% (window start position is not specified.) | R/W |
| b15, b14 | — | Reserved | These bits are read as 0. Writing to these bits has no effect. | R |

There are some restrictions on writing to the IWDTCR register. For details, refer to section 24.3.2, Control over Writing to the IWDTCR, IWDTRCR, and IWDTCSTPR Registers.

In auto-start mode, the settings in the IWDTCR register are disabled, and the settings in option function select register 0 (OFS0) are enabled. The bit setting made to the IWDTCR register can also be made in option function select register 0 (OFS0). For details, refer to section 24.3.8, Correspondence between Option Function Select Register 0 (OFS0) and IWDT Registers.

### TOPS[1:0] Bits (Timeout Period Selection)

The TOPS[1:0] bits select the timeout period (period until the down-counter underflows) from among 128, 512, 1024, or 2048 cycles, taking the divided clock specified by the CKS[3:0] bits as one cycle.
After the down-counter is refreshed, the combination of the CKS[3:0] and TOPS[1:0] bits determines the time (number of IWDTCLK cycles) until the counter underflows.
Relations between the CKS[3:0] and TOPS[1:0] bit setting, the timeout period, and the number of IWDTCLK cycles are listed in Table 24.2.

**Table 24.2 Settings and Timeout Periods**

| CKS[3:0] Bits | | | | TOPS[1:0] Bits | | | | |
|---|---|---|---|---|---|---|---|---|
| b7 | b6 | b5 | b4 | b1 | b0 | Clock Division Ratio | Timeout Period (Number of Cycles) | Cycles of IWDTCLK |
| 0 | 0 | 0 | 0 | 0 | 0 | IWDTCLK | 128 | 128 |
| | | | | 0 | 1 | | 512 | 512 |
| | | | | 1 | 0 | | 1024 | 1024 |
| | | | | 1 | 1 | | 2048 | 2048 |
| 0 | 0 | 1 | 0 | 0 | 0 | IWDTCLK/16 | 128 | 2048 |
| | | | | 0 | 1 | | 512 | 8192 |
| | | | | 1 | 0 | | 1024 | 16384 |
| | | | | 1 | 1 | | 2048 | 32768 |
| 0 | 0 | 1 | 1 | 0 | 0 | IWDTCLK/32 | 128 | 4096 |
| | | | | 0 | 1 | | 512 | 16384 |
| | | | | 1 | 0 | | 1024 | 32768 |
| | | | | 1 | 1 | | 2048 | 65536 |
| 0 | 1 | 0 | 0 | 0 | 0 | IWDTCLK/64 | 128 | 8192 |
| | | | | 0 | 1 | | 512 | 32768 |
| | | | | 1 | 0 | | 1024 | 65536 |
| | | | | 1 | 1 | | 2048 | 131072 |
| 1 | 1 | 1 | 1 | 0 | 0 | IWDTCLK/128 | 128 | 16384 |
| | | | | 0 | 1 | | 512 | 65536 |
| | | | | 1 | 0 | | 1024 | 131072 |
| | | | | 1 | 1 | | 2048 | 262144 |
| 0 | 1 | 0 | 1 | 0 | 0 | IWDTCLK/256 | 128 | 32768 |
| | | | | 0 | 1 | | 512 | 131072 |
| | | | | 1 | 0 | | 1024 | 262144 |
| | | | | 1 | 1 | | 2048 | 524288 |

### CKS[3:0] Bits (Clock Division Ratio Selection)

These bits select the IWDTCLK clock division ratio from among division by 1, 16, 32, 64, 128, and 256. Combination with the TOPS[1:0] bit setting, a count period between 128 and 524288 cycles of the IWDTCLK clock can be selected for the IWDT.

### RPES[1:0] Bits (Window End Position Selection)

These bits select 75%, 50%, 25% or 0% of the count period for the window end position of the down-counter. The window end position should be a value smaller than the window start position (window start position > window end position). If the window end position is greater than the window start position, only the window start position setting is enabled.

The counter values for the window start and end positions selected by setting the RPSS[1:0] and RPES[1:0] bits change depending on the TOPS[1:0] bit setting.

Table 24.3 lists the counter values for the window start and end positions corresponding to TOPS[1:0] bit values.

**Table 24.3 Relationship between Timeout Period and Window Start and End Counter Values**

| TOPS[1:0] Bits | | Timeout Period | | Window Start and End Counter Value | | | |
|---|---|---|---|---|---|---|---|
| b1 | b0 | Cycles | Counter Value | 100% | 75% | 50% | 25% |
| 0 | 0 | 128 | 007Fh | 007Fh | 005Fh | 003Fh | 001Fh |
| 0 | 1 | 512 | 01FFh | 01FFh | 017Fh | 00FFh | 007Fh |
| 1 | 0 | 1024 | 03FFh | 03FFh | 02FFh | 01FFh | 00FFh |
| 1 | 1 | 2048 | 07FFh | 07FFh | 05FFh | 03FFh | 01FFh |

### RPSS[1:0] Bits (Window Start Position Selection)

These bits select a down-counter window start position from 100%, 75%, 50%, or 25% of the count period (100% when the count starts and 0% when the counter underflows). The interval between the window start position and window end position is the refresh-permitted period and the other periods are refresh-prohibited periods.

Figure 24.2 shows the relationship between of the RPSS[1:0] and RPES[1:0] bit setting and the refresh-permitted and refresh-prohibited periods.

| RPSS[1:0] Bits | | RPES[1:0] Bits | | Window | |
|---|---|---|---|---|---|
| b13 | b12 | b9 | b8 | Start (%) | End (%) |
| 1 | 1 | 1 | 1 | 100 | 0 |
| | | 1 | 0 | | 25 |
| | | 0 | 1 | | 50 |
| | | 0 | 0 | | 75 |
| 1 | 0 | 1 | 1 | 75 | 0 |
| | | 1 | 0 | | 25 |
| | | 0 | 1 | | 50 |
| | | 0 | 0 | | 75 |
| 0 | 1 | 1 | 1 | 50 | 0 |
| | | 1 | 0 | | 25 |
| | | 0 | 1 | | 50 |
| | | 0 | 0 | | 75 |
| 0 | 0 | 1 | 1 | 25 | 0 |
| | | 1 | 0 | | 25 |
| | | 0 | 1 | | 50 |
| | | 0 | 0 | | 75 |

Note: • If window end setting ≥ window start setting, the window end setting is set to 0%.

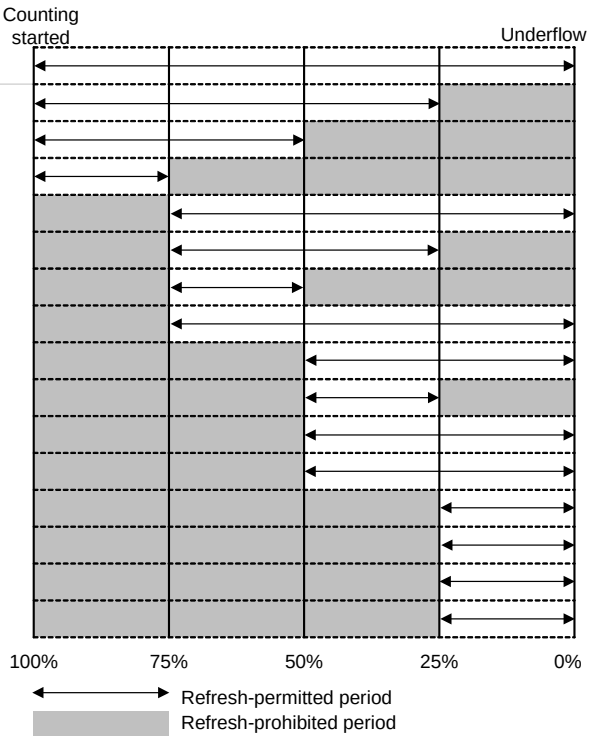



**Figure 24.2 RPSS[1:0] and RPES[1:0] Bit Settings and the Refresh-Permitted Period**

## 24.2.3 IWDT Status Register (IWDTSR)

Address(es): 0008 8034h

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | REFEF | UNDFF | CNTVAL[13:0] | | | | | | | | | | | | | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b13 to b0 | CNTVAL[13:0] | Down-Counter Value | Value counted by the down-counter | R |
| b14 | UNDFF | Underflow Flag | 0: No underflow occurred<br>1: Underflow occurred | R/(W) *1 |
| b15 | REFEF | Refresh Error Flag | 0: No refresh error occurred<br>1: Refresh error occurred | R/(W) *1 |

Note 1. Only 0 can be written to clear the flag.

IWDTSR is initialized by the reset source of the IWDT. IWDTSR is not initialized by other reset sources.

### CNTVAL[13:0] Bits (Down-Counter Value)

Read these bits to confirm the counter value of the down-counter, but note that the read value may differ from the actual count by a value of one count.

### UNDFF Flag (Underflow Flag)

Read this bit can be read to confirm whether or not an underflow has occurred in the down-counter.
The value 1 indicates that the down-counter has underflowed. The value 0 indicates that the down-counter has not underflowed.
Write 0 to the UNDFF flag to set the value to 0. Writing 1 has no effect.

### REFEF Flag (Refresh Error Flag)

Read this bit to confirm whether or not a refresh error (performing a refresh operation during a refresh-prohibited period).
The value 1 indicates that a refresh error has occurred. The value 0 indicates that no refresh error has occurred.
Write 0 to the REFEF flag to set the value to 0. Writing 1 has no effect.

### 24.2.4 IWDT Reset Control Register (IWDTRCR)

Address(es): 0008 8036h

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | RSTIRQS | — | — | — | — | — | — | — |
| Value after reset: | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b6 to b0 | — | Reserved | These bits are read as 0. Writing to these bits has no effect. | R |
| b7 | RSTIRQS | Reset Interrupt Request Selection | 0: Non-maskable interrupt request output is enabled.<br>1: Reset output is enabled. | R/W |

There are some restrictions on writing to the IWDTRCR register. For details, refer to section 24.3.2, Control over Writing to the IWDTCR, IWDTRCR, and IWDTCSTPR Registers.

In auto-start mode, the IWDTRCR register setting are disabled, and the settings in option function select register 0 (OFS0) enabled. The bit setting mode to the IWDTRCR register can also be made in option function select register 0. For details, refer to section 24.3.8, Correspondence between Option Function Select Register 0 (OFS0) and IWDT Registers.

## 24.2.5 IWDT Count Stop Control Register (IWDTCSTPR)

Address(es): 0008 8038h

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | SLCSTP | — | — | — | — | — | — | — |
| Value after reset: | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b6 to b0 | — | Reserved | These bits are read as 0. Writing to these bits has no effect. | R |
| b7 | SLCSTP | Sleep-Mode Count Stop Control | 0: Count stop is disabled.<br>1: Count is stopped at a transition to sleep mode, software standby mode, or deep sleep mode. | R/W |

IWDTCSTPR controls whether to stop the IWDT counter in a low power consumption state. There are some restrictions on writing to the IWDTCSTPR register. For details, refer to section 24.3.2, Control over Writing to the IWDTCR, IWDTRCR, and IWDTCSTPR Registers.
In auto-start mode, the settings in the IWDTCSTPR register are ignored, and the settings in option function select register 0 (OFS0) take effect. The bit setting mode to the IWDTCSTPR register can also be made in option function select register 0 (OFS0). For details, refer to section 24.3.8, Correspondence between Option Function Select Register 0 (OFS0) and IWDT Registers.

**SLCSTP Bit (Sleep-Mode Count Stop Control)**

This bit selects whether to stop counting at a transition to sleep mode, software standby mode, or deep sleep mode.

## 24.2.6 Option Function Select Register 0 (OFS0)

For option function select register 0 (OFS0), refer to section 24.3.8, Correspondence between Option Function Select Register 0 (OFS0) and IWDT Registers.

## 24.3 Operation

### 24.3.1 Count Operation in Each Start Mode

Select the IWDT start mode by setting the IWDT start mode select bit (OFS0.IWDTSTRT) in option function select register 0.
When the OFS0.IWDTSTRT bit is 1 (register start mode), the IWDT control register (IWDTCR), IWDT reset control register (IWDTRCR), and IWDT count stop control register (IWDTCSTPR) are enabled, and counting is started by refreshing (writing) the IWDT refresh register (IWDTRR). When the OFS0.IWDTSTRT bit is 0 (auto-start mode), the setting of option function select register 0 (OFS0) is enabled, and counting automatically starts after reset.

#### 24.3.1.1 Register Start Mode

When the IWDT start mode select (OFS0.IWDTSTRT) bit in option function select register 0 is 1, register start mode is selected, and the IWDT control register (IWDTCR), IWDT reset control register (IWDTRCR), and IWDT count stop control register (IWDTCSTPR) are enabled.
After canceling from the reset, set the clock division ratio, window start and end positions, and timeout period in the IWDTCR register, the reset output or interrupt request output in the IWDTRCR register, and the down-counter stop control at transitions to low power consumption states in the IWDTCSTPR register. Then refresh the down-counter to start counting down from the value selected by setting the timeout period select (IWDTCR.TOPS[1:0]) bits.
There after, as long as the program continues normal operation and the counter is refreshed in the refresh-permitted period, the value in the counter is re-set each time the counter is refreshed and counting down continues. The IWDT does not output the reset signal as long as this continues. However, if the down-counter underflows because of the down-counter cannot be refreshed due to a program runaway, or if a refresh error occurs because the counter was refreshed outside the refresh-permitted period, the IWDT outputs a reset signal or a non-maskable interrupt request (WUNI). Set the IWDT reset interrupt request select bit (OFS0.IWDTRSTIRQS) to select either reset output or interrupt request output.

Figure 24.3 shows an example of operation under the following conditions.

- The IWDT start mode select bit (OFS0.IWDTSTRT) is 1 (register start mode)
- The IWDT reset interrupt request select bit (IWDTRCR.RSTIRQS) is 1 (reset output is enabled)
- The IWDT window start position select bits (IWDTCR.RPSS[1:0]) are 10b (75%)
- The IWDT window end position select bits (IWDTCR.RPES[1:0]) are 10b (25%)

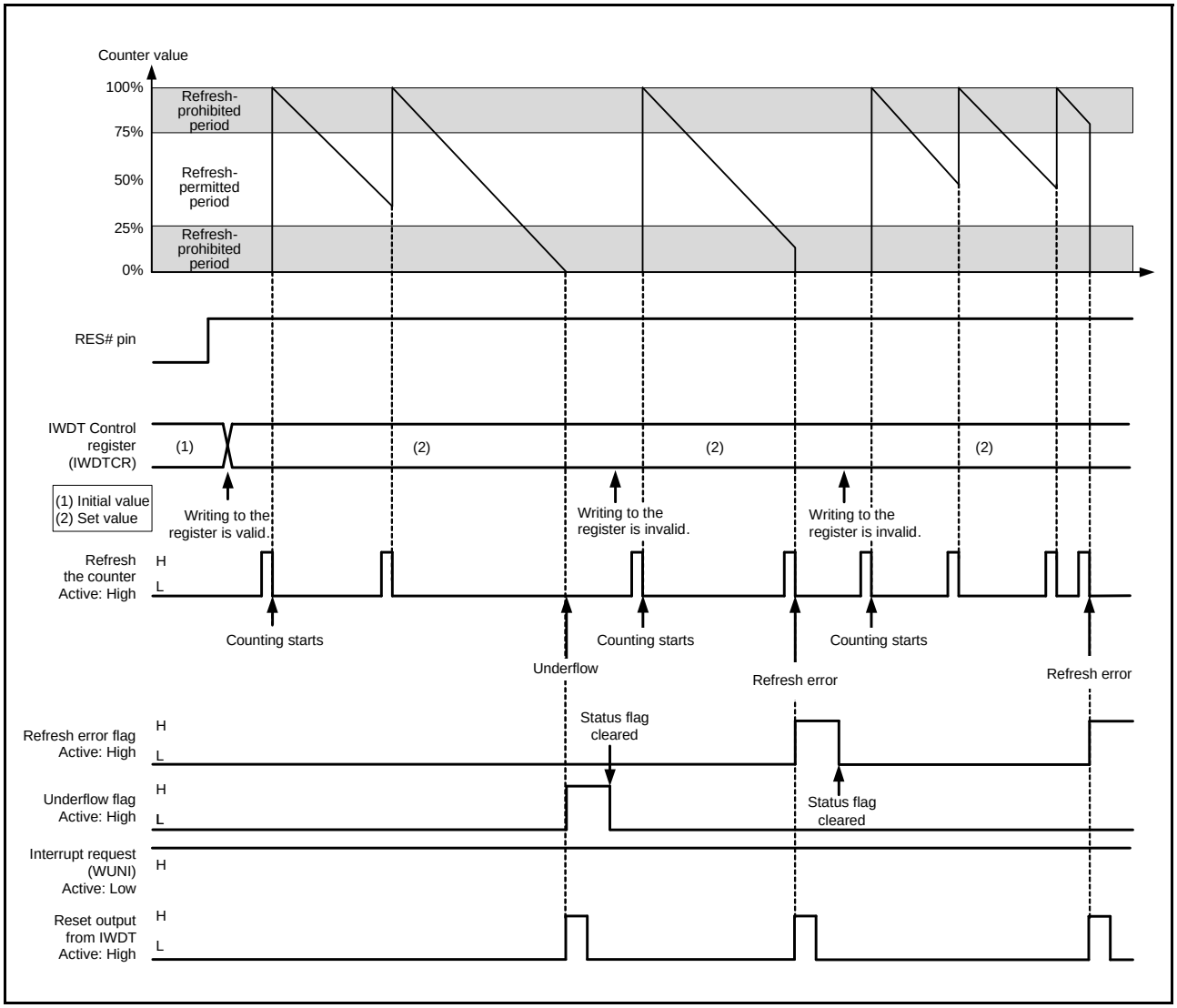



**Figure 24.3 Operation Example in Register Start Mode**

#### 24.3.1.2 Auto-Start Mode

When the IWDT start mode select (OFS0.IWDTSTRT) bit in option function select register 0 is 0, auto-start mode is selected, and the IWDT control register (IWDTCR), IWDT reset control register (IWDTRCR), and IWDT count stop control register (IWDTCSTPR) are disabled.
Within the reset state, the clock division ratio, window start and end positions, timeout period, reset output or interrupt request output, and down-counter stop control at transitions to low power consumption states should be specified in option function select register 0 (OFS0). When the reset state is canceled, the down-counter automatically starts counting down from the value selected by the IWDT timeout period select (OFS0.IWDTTOPS[1:0]) bits.
After that, as long as the program continues normal operation and the counter is refreshed in the refresh-permitted period, the value in the counter is re-set when the counter is refreshed and counting down continues. The IWDT does not output the reset signal as long as this continues. However, if the down-counter underflows because refreshing of the down-counter is not possible due to the program having entered crashed execution or if a refresh error occurs due to refreshing outside the refresh-permitted period, the IWDT outputs the reset signal or non-maskable interrupt request (WUNI). After the reset signal or non-maskable interrupt request is generated, the down-counter reloads the timeout period after counting for one cycle, and restarts counting. Set the IWDT reset interrupt request select bit (OFS0.IWDTRSTIRQS) to select either reset output or interrupt request output.

Figure 24.4 shows an example of operation under the following conditions.

- The IWDT start mode select bit (OFS0.IWDTSTRT) is 0 (auto-start mode)
- The IWDT reset interrupt request select bit (OFS0.IWDTRSTIRQS) is 0 (non-maskable interrupt request output is enabled)
- The IWDT window start position select bits (OFS0.IWDTRPSS[1:0]) are 10b (75%)
- The IWDT window end position select bits (OFS0.IWDTRPES[1:0]) are 10b (25%)

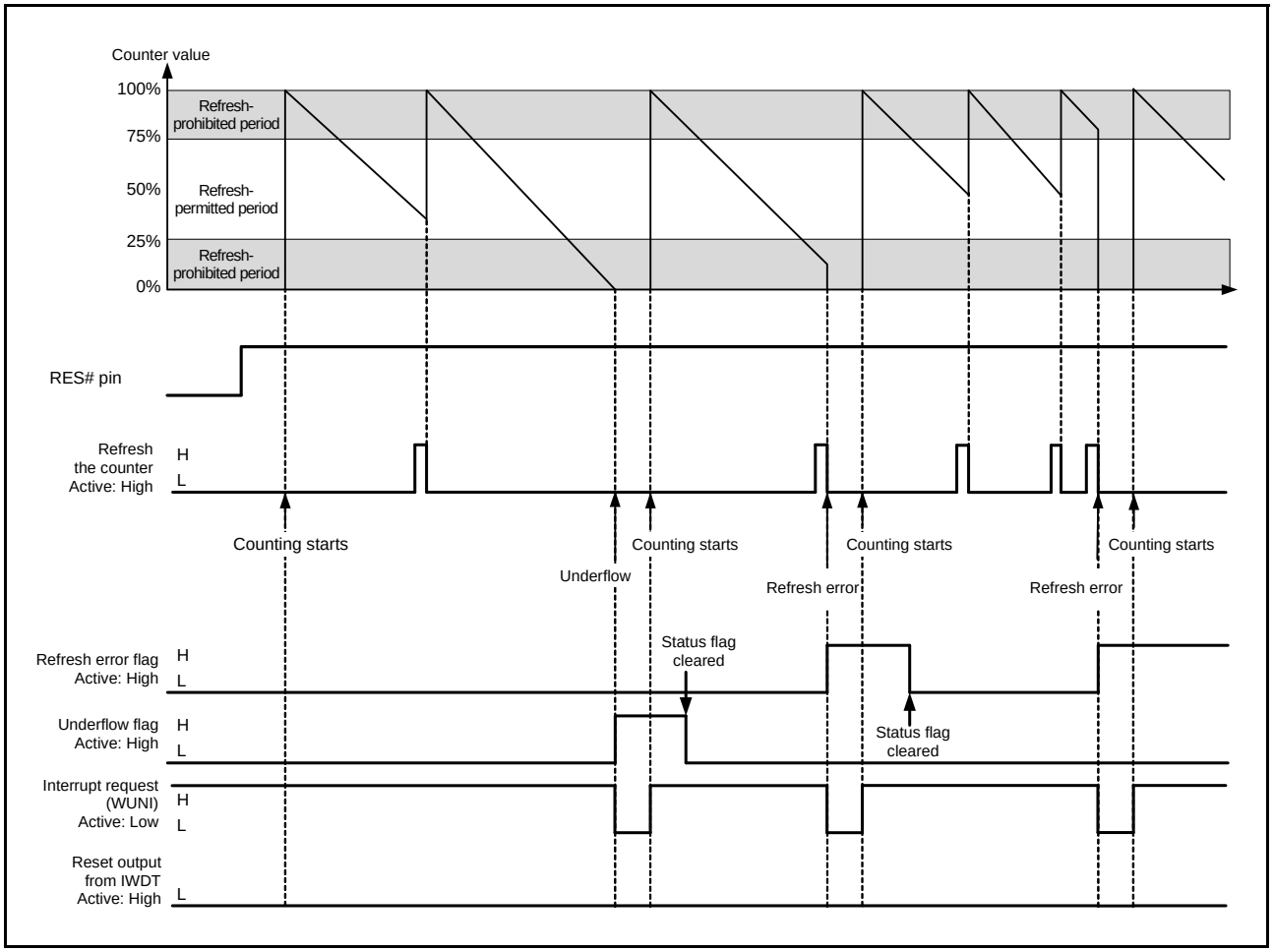



**Figure 24.4 Operation Example in Auto-Start Mode**

### 24.3.2 Control over Writing to the IWDTCR, IWDTRCR, and IWDTCSTPR Registers

Writing to the IWDT control register (IWDTCR), IWDT reset control register (IWDTRCR), or IWDT count stop control register (IWDTCSTPR) is only possible once between the release from the reset state and the first refresh operation. After a refresh operation (counting starts) or IWDTCR, IWDTRCR, or IWDTCSTPR is written to, the protection signal in the IWDT becomes 1 to protect IWDTCR, IWDTRCR, and IWDTCSTPR against subsequent attempts at writing. This protection is released by the reset source of the IWDT. With other reset sources, the protection is not released. Figure 24.5 shows control waveforms produced in response to writing to the IWDTCR register.

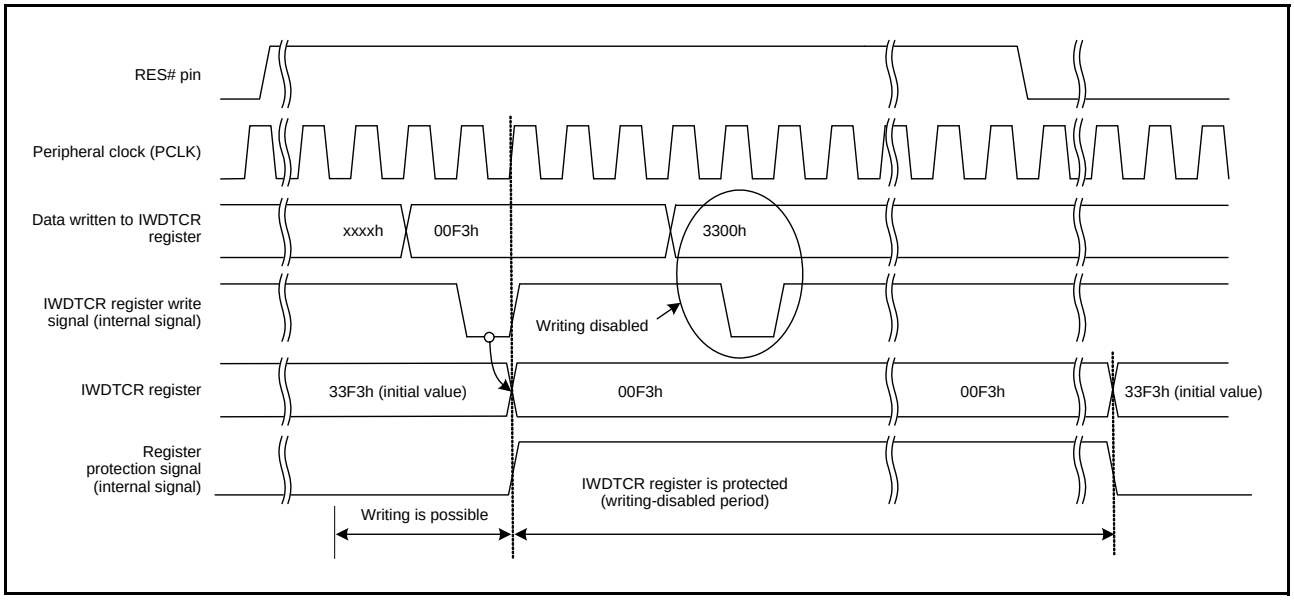



**Figure 24.5 Control Waveforms Produced in Response to Writing to the IWDTCR Register**

### 24.3.3 Refresh Operation

The down-counter is refreshed and starts operation (counting is started by refreshing) by writing the values 00h and then FFh to the IWDT refresh register (IWDTRR). If a value other than FFh is written after 00h, the down-counter is not refreshed. After such invalid writing, correct refreshing is performed by again writing 00h and then FFh to the IWDT refresh register (IWDTRR).
When writing is done in the order of 00h (first time) → 00h (second time), and if FFh is written after that, the writing order 00h → FFh is satisfied; writing 00h (n–1-th time) → 00h (nth time) → FFh is valid and correct refreshing will be done. Even when the first value written before 00h is not 00h, correct refreshing will be done if the operation contains the set of writing 00h → FFh. Moreover, even if a register other than IWDTRR is accessed or IWDTRR is read between writing 00h and writing FFh to IWDTRR, correct refreshing will be done.

[Sample sequences of writing that are valid for refreshing the counter]

- 00h → FFh
- 00h (n–1-th time) → 00h (nth time) → FFh
- 00h → access to another register or read from IWDTRR → FFh

[Sample sequences of writing that are not valid for refreshing the counter]

- 23h (a value other than 00h) → FFh
- 00h → 54h (a value other than FFh)
- 00h → AAh (00h and a value other than FFh) → FFh

Even when 00h is written to IWDTRR outside the refresh-permitted period, if FFh is written to IWDTRR in the refresh-permitted period, the writing sequence is valid and refreshing will be done.
After FFh is written to the IWDTRR register, refreshing the down-counter requires up to four cycles of the signal for counting (the clock division ratio selection (IWDTCR.CKS[3:0]) bits determine how many cycles of the IWDT-dedicated clock (IWDTCLK) make up one cycle for counting). Therefore, writing FFh to the IWDTRR should be completed four-count cycles before the end position of the refresh-permitted period or a counter underflow. The value of the down-counter can be checked by the counter bits (IWDTSR.CNTVAL[13:0]).

[Sample refreshing timings]

- When the window start position is set to 1FFFh, even if 00h is written to IWDTRR before 1FFFh is reached (2002h, for example), refreshing is done if FFh is written to IWDTRR after the value of the IWDTSR.CNTVAL[13:0] bits has reached 1FFFh.
- When the window end position is set to 1FFFh, refreshing is done if 2003h (four-count cycles before 1FFFh) or a greater value is read from the IWDTSR.CNTVAL[13:0] bits immediately after writing 00h → FFh to IWDTRR.
- When the refresh-permitted period continues until count 0000h, refreshing can be done immediately before an underflow. In this case, if 0003h (four-count cycles before an underflow) or a greater value is read from the IWDTSR.CNTVAL[13:0] bits immediately after writing 00h → FFh to IWDTRR, no underflow occurs and refreshing is done.

Figure 24.6 shows the IWDT refresh-operation waveforms when PCLK > IWDTCLK and clock division ratio = IWDTCLK.

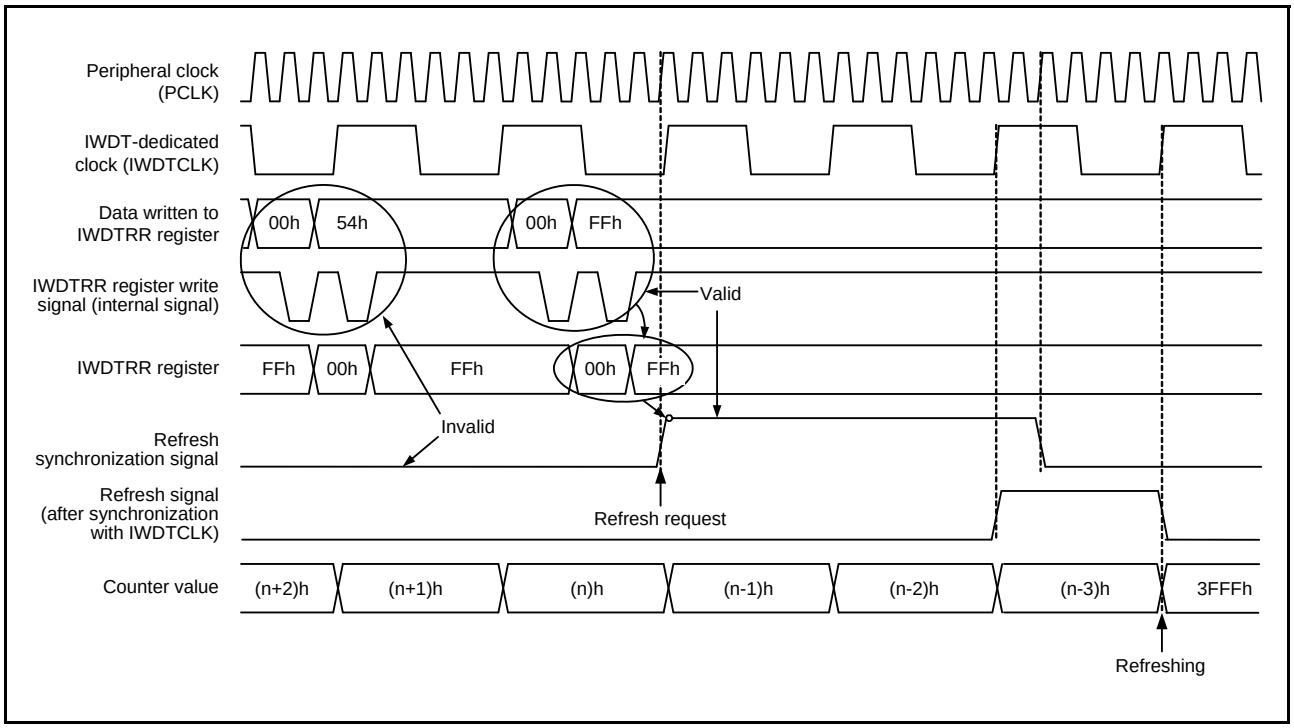



**Figure 24.6 IWDT Refresh Operation Waveforms (IWDTCR.CKS[3:0] = 0000b, IWDTCR.TOPS[1:0] = 11b)**

### 24.3.4 Status Flags

The refresh error (IWDTSR.REFEF) and underflow (IWDTSR.UNDFF) flags retain the source of the reset signal output from the IWDT or the source of the interrupt request from the IWDT.
Thus, after release from the reset state or interrupt request generation, read the IWDTSR.REFEF and IWDTSR.UNDFF flags to check for the reset or interrupt source.
For each flag, writing 0 clears the bit and writing 1 has no effect.
Leaving the status flags unchanged does not affect operation. If the flags are not cleared, at the time of the next reset or interrupt request from the IWDT, the earlier reset or interrupt source is cleared and the new reset or interrupt source is written.
After 0 is written to each flag, up to three IWDTCLK cycles and two PCLK cycles are required before the value is reflected.

### 24.3.5 Reset Output

When the reset interrupt selection (IWDTRCR.RSTIRQS) bit is set to 1 in register start mode or when the IWDT reset interrupt request select (OFS0.IWDTRSTIRQS) bit in option function select register 0 (OFS0) is set to 1 in auto-start mode, a reset signal is output when an underflow in the down-counter or a refresh error occurs.
In register start mode, the down-counter is initialized (all bits cleared to 0) and kept in that state after assertion of the reset signal. After the reset is canceled and the program is restarted, the counter is set up again and counting down is started by refreshing.
In auto-start mode, counting down automatically starts after the reset output.

### 24.3.6 Interrupt Source

When the reset interrupt selection (IWDTRCR.RSTIRQS) bit is set to 0 in register start mode or when the IWDT reset interrupt request select (OFS0.IWDTRSTIRQS) bit in option function select register 0 (OFS0) is set to 0 in auto-start mode, a non-maskable interrupt (WUNI) signal is output when an underflow in the down-counter or a refresh error occurs.

**Table 24.4 IWDT Interrupt Source**

| Name | Interrupt Source | DTC Activation |
|---|---|---|
| WUNI | Down-counter underflow<br>Refresh error | Not possible |

### 24.3.7 Reading the Down-Counter Value

As the down-counter in IWDT-dedicated clock (IWDTCLK), the counter value cannot be read directly. The IWDT synchronizes the counter value with the peripheral clock (PCLK) and stores it in the down-counter value (IWDTSR.CNTVAL[13:0]) bits of the IWDT status register. Thus, the counter value can be checked indirectly through the IWDTSR.CNTVAL[13:0] bits.

Reading the down-counter value requires multiple PCLK clock cycles (up to four clock cycles), and the read counter value may differ from the actual down-counter value by a value of one count.

Figure 24.7 shows the processing for reading the IWDT down-counter value when PCLK > IWDTCLK and clock division ratio = IWDTCLK.

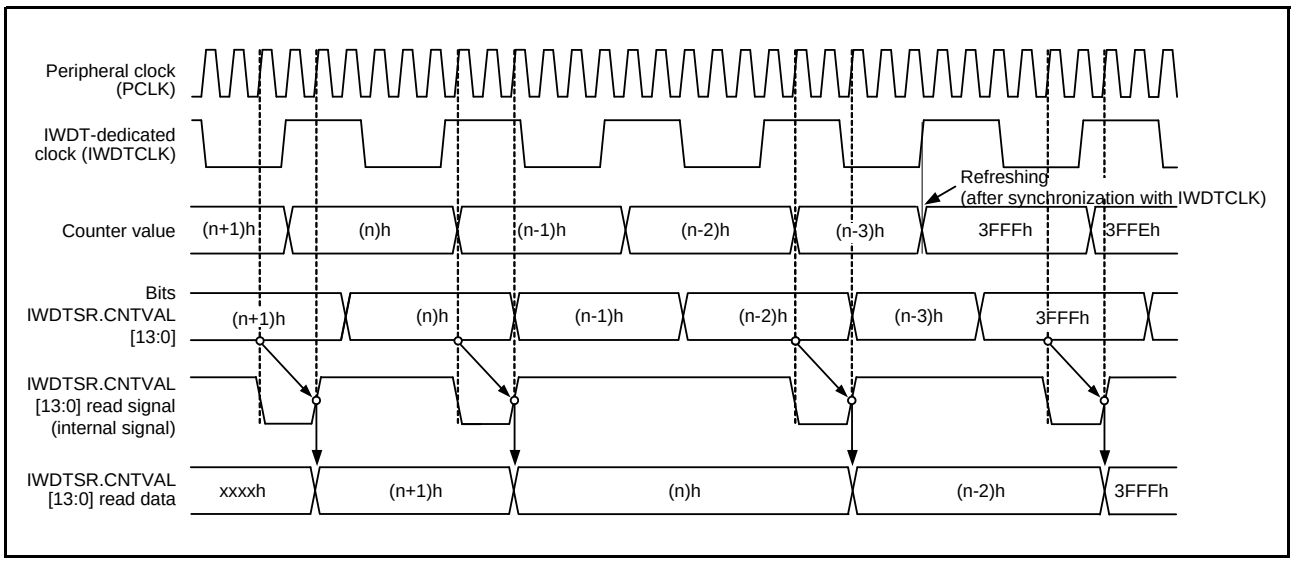



**Figure 24.7 Processing for Reading IWDT Down-Counter Value (IWDTCR.CKS[3:0] = 0000b, IWDTCR.TOPS[1:0] = 11b)**

### 24.3.8 Correspondence between Option Function Select Register 0 (OFS0) and IWDT Registers

Table 24.5 lists the correspondence between option function select register 0 (OFS0) and the IWDT registers (IWDT control register (IWDTCR), IWDT reset control register (IWDTRCR), and IWDT count stop control register (IWDTCSTPR)) regarding control of the down-counter, reset or interrupt request output, and count stop function. Control can be switched between option function select register 0 (OFS0) and the IWDT registers (IWDTCR, IWDTRCR, and IWDTCSTPR) through the setting of the IWDT start mode select (OFS0.IWDTSTRT) bit in option function select register 0 (OFS0).
Note that option function select register 0 (OFS0) setting should be kept unchanged during IWDT operation.
For details on option function select register 0 (OFS0), refer to section 7.2.1, Option Function Select Register 0 (OFS0).

**Table 24.5 Correspondence between Option Function Select Register 0 (OFS0) and IWDT Registers**

| Target of Control | Function | OFS0 Register (Effective in Auto-Start Mode) OFS0.IWDTSTRT = 0 | IWDT Registers (Effective in Register Start Mode) OFS0.IWDTSTRT = 1 |
|---|---|---|---|
| Down-counter | Timeout period selection | OFS0.IWDTTOPS[1:0] | IWDTCR.TOPS[1:0] |
| | Clock division ratio selection | OFS0.IWDTCKS[3:0] | IWDTCR.CKS[3:0] |
| | Window start position selection | OFS0.IWDTRPSS[1:0] | IWDTCR.RPSS[1:0] |
| | Window end position selection | OFS0.IWDTRPES[1:0] | IWDTCR.RPES[1:0] |
| Reset output or interrupt request output | Reset output or interrupt request output selection | OFS0.IWDTRSTIRQS | IWDTRCR.RSTIRQS |
| Count stop | Sleep-mode count stop selection | OFS0.IWDTSLCSTP | IWDTCSTPR.SLCSTP |

## 24.4 Usage Notes

### 24.4.1 Refresh Operations

When making the settings to control the timing of refreshing, consider variations in the range of errors due to the accuracy of the PCLK and IWDTCLK and set values which ensure that refreshing is possible.

### 24.4.2 Clock Division Ratio Setting

Satisfy the frequency of the peripheral module clock (PCLKB) ≥ 4 × (the frequency of the count clock source after division).

# 25. USB 2.0 Host/Function Module (USBc)

## 25.1 Overview

Products of the MCU incorporate a USB2.0 host/function module.
The USB module is a USB controller that is equipped to operate as a host controller or function controller. The module supports full-speed and low-speed transfer as defined in revision 2.0 of the Universal Serial Bus Specification. The module has an internal USB transceiver and supports all of the transfer types defined in USB Specifications 2.0. It has also supports Battery Charging Specification Revision 1.2.
The USB has buffer memory for data transfer, providing a maximum of 10 pipes. Any endpoint numbers can be assigned to PIPE1 to PIPE9, based on the peripheral devices or user system for communication.
Table 25.1 shows the specifications of the USB.

**Table 25.1 USB Specifications**

| Item | Specifications |
|---|---|
| Features | • USB Device Controller (UDC) and transceiver for USB 2.0 are incorporated.<br>Host controller, function controller, and On-The-Go (OTG) are supported (one channel)<br>• The host controller and the function controller can be switched by software.<br>• Self-power mode or bus power mode can be selected.<br>• BC 1.2 (Battery Charging Specification Revision 1.2) is supported. |
| | When host controller operation is selected (only in products with 32 Kbyte or more ROM):<br>• Full-speed transfer (12 Mbps) and low-speed (1.5 Mbps) are supported<br>• Automatic scheduling for SOF and packet transmissions<br>• Programmable intervals for isochronous and interrupt transfers |
| | When function controller operation is selected:<br>• Full-speed transfer (12 Mbps) and low-speed transfer (1.5 Mbps) are supported<br>• Control transfer stage control function<br>• Device state control function<br>• Auto response function for SET_ADDRESS request<br>• SOF interpolation function |
| Communication data transfer type | • Control transfer<br>• Bulk transfer<br>• Interrupt transfer<br>• Isochronous transfer |
| Pipe configuration | • Buffer memory for USB communication is provided.<br>• Up to 10 pipes can be selected (including the default control pipe).<br>• Endpoint numbers can be assigned flexibly to PIPE1 to PIPE9. |
| | Transfer conditions that can be set for each pipe:<br>• PIPE0: Control transfer only (default control pipe: DPC)<br>Buffer size: 8, 16, 32, or 64 bytes (single buffer)<br>• PIPE1 and PIPE2: Bulk transfer or isochronous transfer<br>Buffer size: 8, 16, 32, or 64 bytes for bulk transfer or 1 to 256 bytes for isochronous transfer (double buffer can be specified)<br>• PIPE3 to PIPE5: Bulk transfer only<br>Buffer size: 8, 16, 32, or 64 bytes (double buffer can be specified)<br>• PIPE6 to PIPE9: Interrupt transfer only<br>Buffer size: 1 to 64 bytes (single buffer) |
| Others | • Reception ending function using transaction count<br>• Function that changes the BRDY interrupt event notification timing (BFRE)<br>• Function that automatically clears the buffer memory after the data for the pipe specified at the DnFIFO (n = 0, 1) port has been read (DCLRM)<br>• NAK setting function for response PID generated by end of transfer (SHTNAK)<br>• On-chip pull-up and pull-down resistors of DP/DM |
| Low power consumption function | Module stop state can be set. |

Figure 25.1 shows a block diagram of the USB.

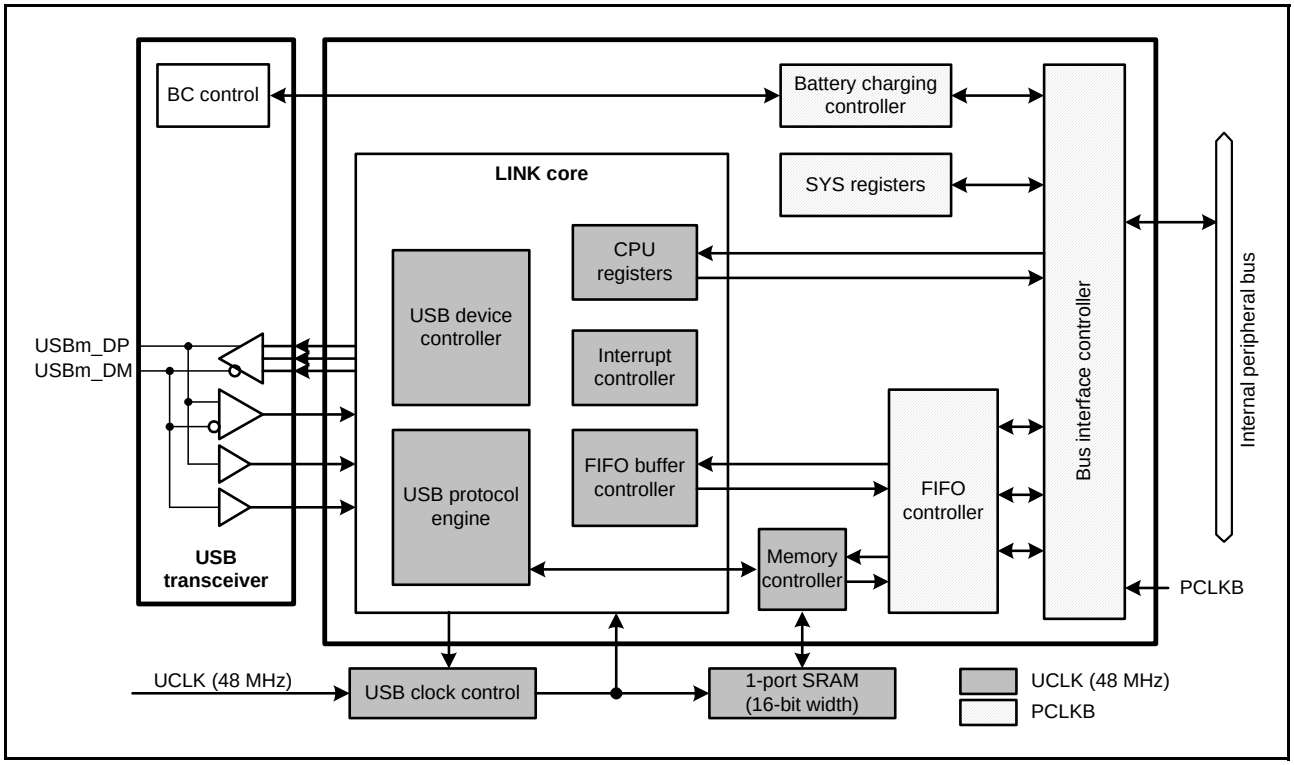



**Figure 25.1 USB Block Diagram**

Table 25.2 lists the I/O pins of the USB.

**Table 25.2 USB Pin Configuration**

| Port | Pin Name | I/O | Function |
|---|---|---|---|
| USB0 | USB0_DP | I/O | D+ I/O pin of the USB on-chip transceiver<br>This pin should be connected to the D+ pin of the USB bus. |
| | USB0_DM | I/O | D– I/O pin of the USB on-chip transceiver<br>This pin should be connected to the D– pin of the USB bus. |
| | USB0_VBUS | Input | USB cable connection monitor pin<br>This pin should be connected to VBUS of the USB bus. Whether VBUS is connected or disconnected can be detected during operation as a function controller. *1 |
| | USB0_EXICEN | Output | Low-power control signal for external power supply (OTG) chip |
| | USB0_VBUSEN | Output | VBUS (5 V) supply enable signal for external power supply chip |
| | USB0_OVRCURA<br>USB0_OVRCURB | Input | External overcurrent detection signals should be connected to these pins. VBUS comparator signals should be connected to these pins when the OTG power supply chip is connected. |
| | USB0_ID | Input | Mini-AB connector ID input signal should be connected to this pin during operation in OTG mode. |
| Common | VCC_USB | Input | USB power supply pin |
| | VSS_USB | Input | USB ground pin |

Note 1. P16 is 5 V tolerant. When using non-5 V tolerant PC4, lower VBUS to 3.3 V and connect to the pin.

## 25.2 Register Descriptions

### 25.2.1 System Configuration Control Register (SYSCFG)

Address(es): 000A 0000h

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | SCKE | — | CNEN | — | DCFM | DRPD | DPRPU | DMRPU | — | — | USBE |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b0 | USBE | USB Operation Enable | 0: USB operation is disabled.<br>1: USB operation is enabled. | R/W |
| b2, b1 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |
| b3 | DMRPU | D– Line Resistor Control *1 | 0: Pulling up the line is disabled.<br>1: Pulling up the line is enabled. | R/W |
| b4 | DPRPU | D+ Line Resistor Control *1 | 0: Pulling up the line is disabled.<br>1: Pulling up the line is enabled. | R/W |
| b5 | DRPD | D+/D– Line Resistor Control | 0: Pulling down the lines is disabled.<br>1: Pulling down the lines is enabled. | R/W |
| b6 | DCFM | Controller Function Select | 0: Function controller is selected.<br>1: Host controller is selected. | R/W |
| b7 | — | Reserved | This bit is read as 0. The write value should be 0. | R/W |
| b8 | CNEN | CNEN Single End Receiver Enable | 0: Single end receiver operation is disabled.<br>1: Single end receiver operation is enabled. | R/W |
| b9 | — | Reserved | This bit is read as 0. The write value should be 0. | R/W |
| b10 | SCKE | USB Clock Enable *2 | 0: Stops supplying the clock signal to the USB.<br>1: Enables supplying the clock signal to the USB. | R/W |
| b15 to b11 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |

Note 1. Do not enable the DMRPU and DPRPU bits at the same time.
Note 2. After writing 1 to the SCKE bit, read this bit and confirm that it is 1.

### USBE Bit (USB Operation Enable)

The USBE bit enables or disables operation of the USB.
Modifying the USBE bit from 1 to 0 initializes some register bits as listed in Table 25.3.
This bit should be modified while the SCKE bit is 1.
When the host controller is selected, this bit should be set to 1 after setting the DRPD bit to 1, eliminating SYSSTS0.LNST[1:0] bit chattering, and checking that the USB bus state has been settled.

**Table 25.3 Registers Initialized by Writing SYSCFG.USBE = 0**

| Selected Function | Register | Bit | Remarks |
|---|---|---|---|
| Function controller | SYSSTS0 | LNST[1:0] | The value is retained when the host controller is selected. |
| | DVSTCTR0 | RHST[2:0] | |
| | INTSTS0 | DVSQ[2:0] | The value is retained when the host controller is selected. |
| | USBREQ | BREQUEST[7:0], BMREQUESTTYPE[7:0] | The value is retained when the host controller is selected. |
| | USBVAL | WVALUE[15:0] | The value is retained when the host controller is selected. |
| | USBINDX | WINDEX[15:0] | The value is retained when the host controller is selected. |
| | USBLENG | WLENGTH[15:0] | The value is retained when the host controller is selected. |
| Host controller | DVSTCTR0 | RHST[2:0] | |
| | FRMNUM | FRNM[10:0] | The value is retained when the function controller is selected. |

### DMRPU Bit (D– Line Resistor Control)

The DMRPU bit enables or disables pulling up the D– line when the function controller is selected.
When the DMRPU bit is set to 1 while the function controller is selected, the bit forces a pull-up of the D– line to notify the USB host of connection as a low-speed device. Modifying the DMRPU bit from 1 to 0 allows the USB to release the D– line, thus notifying the USB host of disconnection.
This bit should be set to 1 if the function controller is selected, and should be set to 0 if the host controller is selected.

### DPRPU Bit (D+ Line Resistor Control)

The DPRPU bit enables or disables pulling up the D+ line when the function controller is selected.
When the DPRPU bit is set to 1 while the function controller is selected, the bit forces a pull-up of the D+ line to notify the USB host of connection as a full-speed device. Modifying the DPRPU bit from 1 to 0 allows the USB to release the D+ line, thus notifying the USB host of disconnection.
This bit should be set to 1 if the function controller is selected, and should be set to 0 if the host controller is selected.

### DRPD Bit (D+/D– Line Resistor Control)

The DRPD bit enables or disables pulling down D+ and D– lines when the host controller is selected.
This bit should be set to 1 if the host controller is selected, and should be set to 0 if the function controller is selected.

#### DCFM Bit (Controller Function Select)

The DCFM bit selects the function of the USB.
This bit should be modified when the DMRPU, DPRPU, and DRPD bits are all 0.

#### CNEN Bit (CNEN Single End Receiver Enable)

Setting the CNEN bit to 1 allows the USB module to enable the single end receiver and set the LNST bit to monitor the status of the D+/D– lines.
The CNEN bit is used when the USB module operates as a portable device for battery charging.

#### SCKE Bit (USB Clock Enable)

The SCKE bit stops or enables supplying 48-MHz clock signals to the USB.
When this bit is 0, only SYSCFG can be read from and written to; the other registers related to the USB cannot be read from or written to.

## 25.2.2 System Configuration Status Register 0 (SYSSTS0)

Address(es): 000A 0004h

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | OVCMON[1:0] | | — | — | — | — | — | — | — | HTACT | — | — | — | IDMON | LNST[1:0] | |
| Value after reset: | 0*1 | 0*1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0*1 | 0 | 0 |

Note 1. Depends on the USB0_OVRCURA/USB0_OVRCURB and USB0_ID pin status.

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b1, b0 | LNST[1:0] | USB Data Line Status Monitor | See Table 25.4. | R |
| b2 | IDMON | External ID0 Input Pin Monitor | 0: USB0_ID pin is low<br>1: USB0_ID pin is high | R |
| b5 to b3 | — | Reserved | These bits are read as 0 and cannot be modified. | R |
| b6 | HTACT | USB Host Sequencer Status Monitor | 0: Host sequencer of the USB is completely stopped.<br>1: Host sequencer of the USB is not completely stopped. | R |
| b13 to b7 | — | Reserved | These bits are read as 0 and cannot be modified. | R |
| b15, b14 | OVCMON[1:0] | External USB0_OVRCURA/ USB0_OVRCURB Input Pin Monitor | The OVCMON[1] bit indicates the status of the USB0_OVRCURA pin.<br>The OVCMON[0] bit indicates the status of the USB0_OVRCURB pin. | R |

### LNST[1:0] Bits (USB Data Line Status Monitor)

The LNST[1:0] bits indicate the state of the USB data lines (D+ and D– lines). See Table 25.4.

The LNST[1:0] bits should be read after the connection processing (SYSCFG.DPRPU bit = 1 is set) when the function controller is selected; whereas after enabling pull-down of the lines (SYSCFG.DRPD bit = 1 is set) when the host controller is selected.

### HTACT Bit (USB Host Sequencer Status Monitor)

The HTACT bit is 0 when the host sequencer of the USB is completely stopped.

Make sure the HTACT bit is 0 when stopping the clock supply to the USB.

### OVCMON[1:0] Bits (External USB0_OVRCURA/USB0_OVRCURB Input Pin Monitor)

The OCVMON[1:0] bits indicate the status of overcurrent from an external power-supply chip.

**Table 25.4 Status of USB Data Bus Lines (D+ Line, D– Line)**

| LNST[1:0] Bits | During Low-Speed Operation (only when the host controller is selected) | During Full-Speed Operation |
|---|---|---|
| 00b | SE0 | SE0 |
| 01b | K-State | J-State |
| 10b | J-State | K-State |
| 11b | SE1 | SE1 |

## 25.2.3 Device State Control Register 0 (DVSTCTR0)

Address(es): 000A 0008h

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | HNPBTOA | EXICEN | VBUSEN | WKUP | RWUPE | USBRST | RESUME | UACT | — | RHST[2:0] | | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b2 to b0 | RHST[2:0] | USB Bus Reset Status | • When the host controller is selected<br>b2 b0<br>0 0 0: Communication speed not determined (powered state or no connection)<br>1 x x: USB bus reset in progress<br>0 0 1: Low-speed connection<br>0 1 0: Full-speed connection<br><br>x: Don't care<br><br>• When the function controller is selected<br>b2 b0<br>0 0 0: Communication speed not determined<br>0 0 1: USB bus reset in progress or low-speed connection<br>0 1 0: USB bus reset in progress or full-speed connection | R |
| b3 | — | Reserved | This bit is read as 0. The write value should be 0. | R/W |
| b4 | UACT | USB Bus Enable | 0: Downstream port is disabled (SOF transmission is disabled).<br>1: Downstream port is enabled (SOF transmission is enabled). | R/W |
| b5 | RESUME | Resume Output | 0: Resume signal is not output.<br>1: Resume signal is output. | R/W |
| b6 | USBRST | USB Bus Reset Output | 0: USB bus reset signal is not output.<br>1: USB bus reset signal is output. | R/W |
| b7 | RWUPE | Wakeup Detection Enable | 0: Downstream port wakeup is disabled.<br>1: Downstream port wakeup is enabled. | R/W |
| b8 | WKUP | Wakeup Output | 0: Remote wakeup signal is not output.<br>1: Remote wakeup signal is output. | R/W *1 |
| b9 | VBUSEN | USB0_VBUSEN Output Pin Control | 0: External USB0_VBUSEN pin outputs low<br>1: External USB0_VBUSEN pin outputs high | R/W |
| b10 | EXICEN | USB0_EXICEN Output Pin Control | 0: External USB0_EXICEN pin outputs low<br>1: External USB0_EXICEN pin outputs high | R/W |
| b11 | HNPBTOA | Host Negotiation Protocol (HNP) Control | This bit is used when switching from device B to device A while in OTG mode. If the HNPBTOA bit is 1, the internal function control keeps the suspended state until the HNP processing ends even though SYSCFG.DPRPU = 0 or SYSCFG.DCFM = 1 is set. | R/W |
| b15 to b12 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |

Note 1. Only 1 can be written.

### RHST[2:0] Bits (USB Bus Reset Status)

The RHST[2:0] bits indicate the status of the USB bus reset.
When the host controller is selected, the RHST[2:0] bits indicate 100b after the USBRST bit has been set to 1 by software.
The USB fixes the value of the RHST[2:0] bits when the USBRST bit is written to 0 by software and the USB completes SE0 driving.
When the function controller is selected, the RHST[2:0] bits indicate 010b (connection while DPRPU = 1) or 001b (disconnection while DMRPU = 1) when the USB detects the USB bus reset, and a DVST interrupt is generated.

### UACT Bit (USB Bus Enable)

The UACT bit enables operation of the USB bus (controls the SOF packet transmission to the USB bus) when the host controller is selected.
With this bit set to 1, the USB puts the USB port to the USB-bus enabled state and performs SOF output and data transmission and reception.
This module starts outputting SOF packets within one frame after the UACT bit has been written to 1 by software.
With this bit set to 0, the USB enters the idle state after outputting SOF packets.
The USB sets the UACT bit to 0 on any of the following conditions.

- A DTCH interrupt is detected during communication (while UACT = 1).
- An EOFERR interrupt is detected during communication (while UACT = 1).

Writing 1 to this bit should be done at the end of the USB reset processing (writing 0 to the USBRST bit) or at the end of the resume processing from the suspended state (writing 0 to the RESUME bit).
This bit should be set to 0 if the function controller is selected.

### RESUME Bit (Resume Output)

The RESUME bit controls the resume signal output when the host controller is selected.
Setting the RESUME bit to 1 allows the USB to drive the port to the K-state and output the resume signal.
The USB sets the RESUME bit to 1 on detecting the remote wakeup signal while RWUPE is 1 in the USB suspended state.
The USB continues outputting K-state while the RESUME bit = 1 (until the RESUME bit is set to 0 by software). The RESUME bit should be 1 (= resume period) for the time defined by USB Specifications 2.0.
This bit should be set to 1 in the suspended state.
Write 1 to the UACT bit simultaneously with the end of the resume processing (writing 0 to the RESUME bit).
This bit should be set to 0 if the function controller is selected.

### USBRST Bit (USB Bus Reset Output)

The USBRST bit controls the USB bus reset signal output when the host controller is selected.
When the host controller is selected, setting this bit to 1 allows the USB to drive SE0 of the USB port to reset the USB bus.
The USB continues outputting SE0 while USBRST = 1 (until the USBRST bit is set to 1 by software). The USBRST bit should be 1 (= USB bus reset period) for the time defined by USB Specifications 2.0.
Writing 1 to this bit during communication (the UACT bit = 1) or during the resume processing (the RESUME bit = 1) prevents the USB from starting the USB bus reset processing until both the UACT and RESUME bits become 0.
Write 1 to the UACT bit simultaneously with the end of the USB bus reset processing (writing 0 to the USBRST bit).
This bit should be set to 0 if the function controller is selected.

### RWUPE Bit (Wakeup Detection Enable)

The RWUPE bit enables or disables the downstream port peripheral device to use the remote wakeup function (resume signal output) when the host controller is selected.
With this bit set to 1, on detecting the remote wakeup signal, the USB detects the resume signal (K-state for 2.5 μs) from the downstream port device and performs the resume processing (drives the port to the K-state).
With this bit set to 0, the USB ignores the detected remote wakeup signal (K-state) from the peripheral device connected to the USB port.
While the PWUPE bit is 1, the internal clock should not be stopped even in the suspended state (SYSCFG.SCKE bit should be set to 1).
This bit should be set to 0 if the function controller is selected.

### WKUP Bit (Wakeup Output)

The WKUP bit enables or disables outputting the remote wakeup signal (resume signal) to the USB bus when the function controller is selected.
The USB controls the output time of a remote wakeup signal. When this bit is set to 1, the USB clears this bit to 0 after outputting the 10-ms K-state.
According to USB Specifications 2.0, the USB bus idle state must be kept for 5 ms or longer before a remote wakeup signal is sent. If the USB writes 1 to this bit right after detection of the suspended state, the K-state will be output after 2 ms.
Do not write 1 to this bit, unless the device state is in the suspended state (bits INTSTS0.DVSQ[2:0] = 1xxb) and the USB host enables the remote wakeup signal. When this bit is set to 1, the internal clock must not be stopped even in the suspended state (write 1 to this bit while the SYSCFG.SCKE bit = 1).
This bit should be set to 0 if the host controller is selected.

### HNPBTOA Bit (Host Negotiation Protocol (HNP) Control)

The HNPBTOA bit is used when switching from device B to device A while in OTG mode. If the HNPBTOA bit is 1, the internal function control keeps the suspended state until the HNP processing ends even though the SYSCFG.DPRPU bit = 0 or SYSCFG.DCFM = 1 is set. Even if the falling edge of the D+ signal is detected at this time, no resume (RESM) interrupt is generated.
After this bit is set to 1, write 0 to this bit by software to terminate the HNP processing when connection to the host (pull-up on the target side) or timeout of the HNP processing is detected.

## 25.2.4 CFIFO Port Register (CFIFO)
## D0FIFO Port Register (D0FIFO)
## D1FIFO Port Register (D1FIFO)

Address(es): CFIFO: 000A 0014h, D0FIFO: 000A 0018h, D1FIFO: 000A 001Ch

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | FIFOPORT[15:0] | | | | | | | | | | | | | | | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b15 to b0 | FIFOPORT[15:0] | FIFO Port | The valid bits in a FIFO port register depend on the settings of the MBW bits (CFIFOSEL.MBW, D0FIFOSEL.MBW, and D1FIFOSEL.MBW) and BIGEND bits (CFIFOSEL.BIGEND, D0FIFOSEL.BIGEND, and D1FIFOSEL.BIGEND) as shown in Table 25.5 and Table 25.6. | R/W |

There are three FIFO ports: CFIFO, D0FIFO, and D1FIFO ports. Each FIFO port is configured of a port register (CFIFO, D0FIFO, or D1FIFO) that handles reading of data from the FIFO buffer memory and writing of data to the FIFO buffer memory, a port select register (CFIFOSEL, D0FIFOSEL, or D1FIFOSEL) that is used to select the pipe assigned to the FIFO port, and a port control register (CFIFOCTR, D0FIFOCTR, or D1FIFOCTR).
Each FIFO port has the following features.

- The FIFO buffer for DCP (control transfer) should be accessed through the CFIFO port.
- Accessing the FIFO buffer using DTC transfer should be performed through the D0FIFO or D1FIFO port.
- The D1FIFO and D0FIFO ports can be accessed also by the CPU.
- When using functions specific to the FIFO port, the pipe number (selected pipe) specified by the CURPIPE[3:0] bits in the port select register cannot be changed (when the DTC transfer function is used, etc.).
- Registers configuring a FIFO port do not affect other FIFO ports.
- The same pipe should not be assigned to two or more FIFO ports.
- There are two FIFO buffer states: the access right is on the CPU side and it is on the Serial Interface Engine (SIE) side. When the FIFO buffer access right is on the SIE side, the FIFO buffer cannot be accessed from the CPU.

### FIFOPORT[15:0] Bits (FIFO Port)

Accessing the FIFOPORT[15:0] bits allow reading the received data from the FIFO buffer or writing the transmit data to the FIFO buffer.

Each FIFO port register can be accessed only while the FRDY bit in each port control register (CFIFOSEL, D0FIFOSEL, or D1FIFOSEL) is 1.

The valid bits in a FIFO port register depend on the settings of the corresponding MBW and BIGEND bits in a port select register (CFIFOSEL, D0FIFOSEL, or D1FIFOSEL) as shown in Table 25.5 and Table 25.6.

**Table 25.5 Endian Operation in 16-Bit Access**

| CFIFOSEL.BIGEND Bit<br>D0FIFOSEL.BIGEND Bit<br>D1FIFOSEL.BIGEND Bit | Bits 15 to 8 | Bits 7 to 0 |
|---|---|---|
| 0 | N + 1 data | N + 0 data |
| 1 | N + 0 data | N + 1 data |

**Table 25.6 Endian Operation in 8-Bit Access**

| CFIFOSEL.BIGEND Bit<br>D0FIFOSEL.BIGEND Bit<br>D1FIFOSEL.BIGEND Bit | Bits 15 to 8 | Bits 7 to 0 |
|---|---|---|
| 0 | Access prohibited*1 | N + 0 data |
| 1 | Access prohibited*1 | N + 0 data |

Note 1. Reading from an access-prohibited area is not allowed.

## 25.2.5 CFIFO Port Select Register (CFIFOSEL) D0FIFO Port Select Register (D0FIFOSEL) D1FIFO Port Select Register (D1FIFOSEL)

- CFIFOSEL

Address(es): 000A 0020h

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | RCNT | REW | — | — | — | MBW | — | BIGEND | — | — | ISEL | — | CURPIPE[3:0] | | | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b3 to b0 | CURPIPE [3:0] | CFIFO Port Access Pipe Specification | b3 b0<br>0 0 0 0: DCP (Default control pipe)<br>0 0 0 1: Pipe 1<br>0 0 1 0: Pipe 2<br>0 0 1 1: Pipe 3<br>0 1 0 0: Pipe 4<br>0 1 0 1: Pipe 5<br>0 1 1 0: Pipe 6<br>0 1 1 1: Pipe 7<br>1 0 0 0: Pipe 8<br>1 0 0 1: Pipe 9<br>Other than above: Setting prohibited | R/W |
| b4 | — | Reserved | This bit is read as 0. The write value should be 0. | R/W |
| b5 | ISEL | CFIFO Port Access Direction When DCP is Selected | 0: Reading from the buffer memory is selected<br>1: Writing to the buffer memory is selected | R/W |
| b7, b6 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |
| b8 | BIGEND | CFIFO Port Endian Control | 0: Little endian<br>1: Big endian | R/W |
| b9 | — | Reserved | This bit is read as 0. The write value should be 0. | R/W |
| b10 | MBW | CFIFO Port Access Bit Width | 0: 8-bit width<br>1: 16-bit width | R/W |
| b13 to b11 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |
| b14 | REW | Buffer Pointer Rewind | 0: The buffer pointer is not rewound.<br>1: The buffer pointer is rewound. | R/W*1 |
| b15 | RCNT | Read Count Mode | 0: The DTLN[8:0] bits (CFIFOSEL.DTLN[8:0], D0FIFOSEL.DTLN[8:0], D1FIFOSEL.DTLN[8:0]) are cleared when all of the receive data has been read from the CFIFO. (In double buffer mode, the DTLN[8:0] bit value is cleared when all the data has been read from only a single plane.)<br>1: The DTLN[8:0] bits are decremented each time the receive data is read from the CFIFO. | R/W |

Note 1. Only 0 can be read.

The same pipe should not be specified by the CURPIPE[3:0] bits in the CFIFOSEL, D0FIFOSEL, and D1FIFOSEL registers. When the CURPIPE[3:0] bits in the D0FIFOSEL and D1FIFOSEL registers are cleared to 0000b, no pipe is selected.

The pipe number should not be changed while the DTC transfer is enabled.

#### CURPIPE[3:0] Bits (CFIFO Port Access Pipe Specification)

The CURPIPE[3:0] bits specify the pipe number using which data is read or written through the CFIFO port.
After writing to these bits, read these bits to check that the written value agrees with the read value before proceeding to the next process.
Do not set the same pipe number to the CURPIPE[3:0] bits in CFIFOSEL, D0FIFOSEL, and D1FIFOSEL.
Even if an attempt is made to modify the setting of these bits during access to the FIFO buffer, the current access setting is retained until the access is completed. Then, the modification becomes effective, thus enabling continuous access.

#### ISEL Bit (CFIFO Port Access Direction When DCP is Selected)

After writing to the ISEL bit with the DCP being a selected pipe, read this bit to check that the written value agrees with the read value before proceeding to the next process.
Set this bit and the CURPIPE[3:0] bits simultaneously.

#### MBW Bit (CFIFO Port Access Bit Width)

The MBW bit specifies the bit width for accessing the CFIFO port.
When the selected pipe is in the receiving direction, once reading data is started after setting this bit, this bit should not be modified until all the data has been read.
When the selected pipe is in the receiving direction, set the CURPIPE[3:0] bits and MBW bits simultaneously.
When the selected pipe is in the transmitting direction, the bit width cannot be changed from 8-bit width to 16-bit width while data is being written to the buffer memory.
An odd number of bytes can also be written through byte-access control even when 16-bit width is selected.

#### REW Bit (Buffer Pointer Rewind)

The REW bit specifies whether or not to rewind the buffer pointer.
When the selected pipe is in the receiving direction, setting the REW bit to 1 while the FIFO buffer is being read allows re-reading the FIFO buffer from the first data (in double buffer mode, re-reading the currently-read FIFO buffer plane from the first data is allowed).
Do not set the REW bit to 1 simultaneously with modifying the CURPIPE[3:0] bits. Before setting the REW bit to 1, be sure to check that the FRDY bit is 1.
To re-write to the FIFO buffer again from the first data for the pipe in the transmitting direction, use the BCLR bit.

- D0FIFOSEL, D1FIFOSEL

Address(es): D0FIFOSEL: 000A 0028h, D1FIFOSEL: 000A 002Ch

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | RCNT | REW | DCLRM | DREQE | — | MBW | — | BIGEND | — | — | — | — | CURPIPE[3:0] | | | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b3 to b0 | CURPIPE [3:0] | FIFO Port Access Pipe Specification | b3 b0<br>0 0 0 0: DCP (Default control pipe)<br>0 0 0 1: Pipe 1<br>0 0 1 0: Pipe 2<br>0 0 1 1: Pipe 3<br>0 1 0 0: Pipe 4<br>0 1 0 1: Pipe 5<br>0 1 1 0: Pipe 6<br>0 1 1 1: Pipe 7<br>1 0 0 0: Pipe 8<br>1 0 0 1: Pipe 9<br>Other than above: Setting prohibited | R/W |
| b7 to b4 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |
| b8 | BIGEND | FIFO Port Endian Control | 0: Little endian<br>1: Big endian | R/W |
| b9 | — | Reserved | This bit is read as 0. The write value should be 0. | R/W |
| b10 | MBW | FIFO Port Access Bit Width | 0: 8-bit width<br>1: 16-bit width | R/W |
| b11 | — | Reserved | This bit is read as 0. The write value should be 0. | R/W |
| b12 | DREQE | DTC Transfer Request Enable | 0: DTC transfer request is disabled.<br>1: DTC transfer request is enabled. | R/W |
| b13 | DCLRM | Auto Buffer Memory Clear Mode Accessed after Specified Pipe Data is Read | 0: Auto buffer clear mode is disabled.<br>1: Auto buffer clear mode is enabled. | R/W |
| b14 | REW | Buffer Pointer Rewind | 0: The buffer pointer is not rewound.<br>1: The buffer pointer is rewound. | R/W*1 |
| b15 | RCNT | Read Count Mode | 0: The DTLN[8:0] bits (CFIFOSEL.DTLN[8:0], D0FIFOSEL.DTLN[8:0], D1FIFOSEL.DTLN[8:0]) are cleared when all of the receive data has been read from the DnFIFO. (In double buffer mode, the DTLN bit Value is cleared when all the data has been read from only a single plane.)<br>1: The DTLN[8:0] bits are decremented each time the receive data is read from the DnFIFO. (n = 0, 1) | R/W |

Note 1. Only 0 can be read.

The same pipe should not be specified by the CURPIPE[3:0] bits in the CFIFOSEL, D0FIFOSEL, and D1FIFOSEL registers. When the CURPIPE[3:0] bits in the D0FIFOSEL and D1FIFOSEL registers are cleared to 0000b, no pipe is selected.

The pipe number should not be changed while the DTC transfer is enabled.

### CURPIPE[3:0] Bits (FIFO Port Access Pipe Specification)

The CURPIPE[3:0] bits specify the pipe number using which data is read or written through the D0FIFO port or D1FIFO port.
After writing to these bits, read these bits to check that the written value agrees with the read value before proceeding to the next process.
Do not set the same pipe number to the CURPIPE[3:0] bits in CFIFOSEL, D0FIFOSEL, and D1FIFOSEL.
Even if an attempt is made to modify the setting of these bits during access to the FIFO buffer, the current access setting is retained until the access is completed. Then, the modification becomes effective, thus enabling continuous access.

### MBW Bit (FIFO Port Access Bit Width)

The MBW bit specifies the bit width for accessing the D0FIFO port or D1FIFO port.
When the selected pipe is in the receiving direction, once reading data is started after setting this bit, this bit should not be modified until all the data has been read.
When the selected pipe is in the receiving direction, set the CURPIPE[3:0] bits and MBW bits simultaneously.
When the selected pipe is in the transmitting direction, the bit width cannot be changed from 8-bit width to 16-bit width while data is being written to the buffer memory.
An odd number of bytes can also be written through byte-access control even when 16-bit width is selected.

### DREQE Bit (DTC Transfer Request Enable)

The DREQE bit enables or disables the DTC transfer request to be issued.
Before setting the DREQE bit to 1 to enable the DTC transfer request to be issued, set the CURPIPE[3:0] bits.
When modifying the setting of the CURPIPE[3:0] bits, set this bit to 0 first.

### DCLRM Bit (Auto Buffer Memory Clear Mode Accessed after Specified Pipe Data is Read)

The DCLRM bit enables or disables the buffer memory to be cleared automatically after data has been read using the selected pipe.
With this bit set to 1, the USB sets the BCLR bit to 1 for the FIFO buffer of the selected pipe on receiving a zero-length packet while the FIFO buffer assigned to the selected pipe is empty, or on receiving a short packet and reading the data while the PIPECFG.BFRE bit is 1.
When using the USB with the SOFCFG.BRDYM bit set to 1, set this bit to 0.

### REW Bit (Buffer Pointer Rewind)

The REW bit specifies whether or not to rewind the buffer pointer.
When the selected pipe is in the receiving direction, setting the REW bit to 1 while the FIFO buffer is being read allows re-reading the FIFO buffer from the first data (in double buffer mode, re-reading the currently-read FIFO buffer plane from the first data is allowed).
Do not set the REW bit to 1 simultaneously with modifying the CURPIPE[3:0] bits. Before setting the REW bit to 1, be sure to check that the FRDY bit is 1.
To re-write to the FIFO buffer again from the first data for the pipe in the transmitting direction, use the BCLR bit.

### RCNT Bit (Read Count Mode)

The RCNT bit specifies the read mode for the value in the CFIFOCTR.DTLN bit.
When accessing DnFIFO with the PIPECFG.BFRE bit set to 1, set the RCNT bit to 0.

## 25.2.6 CFIFO Port Control Register (CFIFOCTR)
## D0FIFO Port Control Register (D0FIFOCTR)
## D1FIFO Port Control Register (D1FIFOCTR)

Address(es): CFIFOCTR: 000A 0022h, D0FIFOCTR: 000A 002Ah, D1FIFOCTR: 000A 002Eh

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | BVAL | BCLR | FRDY | — | — | — | — | DTLN[8:0] | | | | | | | | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b8 to b0 | DTLN[8:0] | Receive Data Length | Indicates the length of the receive data. These bits indicate different values depending on the setting of the RCNT bit in the port select register. For details, refer to the description on the DTLN[8:0] bits shown below. | R |
| b12 to b9 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |
| b13 | FRDY | FIFO Port Ready | 0: FIFO port access is disabled.<br>1: FIFO port access is enabled. | R |
| b14 | BCLR | CPU Buffer Clear | 0: Does not operate.<br>1: Clears the buffer memory on the CPU side. | R/W *1 |
| b15 | BVAL | Buffer Memory Valid Flag | 0: Invalid<br>1: Writing ended | R/W *2 |

Note 1. Only 0 can be read.
Note 2. Only 1 can be written.

CFIFOCTR, D0FIFOCTR, and D1FIFOCTR correspond to CFIFO, D0FIFO, and D1FIFO, respectively.

**DTLN[8:0] Bits (Receive Data Length)**

The DTLN[8:0] bits indicate the length of the receive data.
While the FIFO buffer is being read, the DTLN[8:0] bits indicate different values depending on the DnFIFOSEL.RCNT (n = 0, 1) bit value as described below.

- RCNT = 0
  The USB sets the DTLN[8:0] bits to indicate the length of the receive data until the CPU or DTC has read all the received data from a single FIFO buffer plane.
  While the PIPECFG.BFRE bit = 1, these bits retain the length of the receive data until the BCLR bit is set to 1 even after all the data has been read.
- RCNT = 1
  The USB decrements the value indicated by the DTLN[8:0] bits each time data is read from the FIFO buffer. (The value is decremented by one when the MBW bit is 0, and by two when the MBW bit is 1.)
  The USB sets these bits to 0 when all the data has been read from one FIFO buffer plane. However, in double buffer mode, if data has been received in one FIFO buffer plane before all the data has been read from the other plane, the USB sets these bits to indicate the length of the receive data in the former plane when all the data has been read from the latter plane.

### FRDY Bit (FIFO Port Ready)

The FRDY bit indicates whether the FIFO port can be accessed by the CPU or DTC.
In the following cases, the USB sets the FRDY bit to 1 but data cannot be read via the FIFO port because there is no data to be read. In these cases, set the BCLR bit to 1 to clear the FIFO buffer, and enable transmission and reception of the next data.

- A zero-length packet is received when the FIFO buffer assigned to the selected pipe is empty.
- A short packet is received and the data is completely read while the PIPECFG.BFRE bit = 1.

### BCLR Bit (CPU Buffer Clear)

The BCLR bit should be set to 1 to clear the FIFO buffer on the CPU side for the selected pipe.
When double buffer mode is set for the FIFO buffer assigned to the selected pipe, the USB clears only one plane of the FIFO buffer even when both planes are read-enabled.
When the selected pipe is the DCP, setting the BCLR bit to 1 allows the USB to clear the FIFO buffer regardless of whether the FIFO buffer is on the CPU side or SIE side. To clear the buffer on the SIE side, set the DCPCTR.PID[1:0] bits for the DCP to NAK before setting the BCLR bit to 1.
When the selected pipe is in the transmitting direction, if 1 is written to the BVAL flag and the BCLR bit simultaneously, the USB clears the data that has been written before it, enabling transmission of a zero-length packet.
When the selected pipe is not the DCP, writing 1 to the BCLR bit should be done while the FRDY bit in the FIFO port control register is 1 (set by the USB).

### BVAL Flag (Buffer Memory Valid Flag)

The BVAL flag should be set to 1 when data has been completely written to the FIFO buffer on the CPU side for the pipe selected using the CURPIPE[3:0] bits (selected pipe).
When the selected pipe is in the transmitting direction, set the BVAL flag to 1 in the following cases. Then, the USB switches the FIFO buffer from the CPU side to the SIE side, thus enabling transmission.

- To transmit a short packet, set the BVAL flag to 1 after data has been written.
- To transmit a zero-length packet, set the BVAL flag to 1 before data is written to the FIFO buffer.

When data of the maximum packet size has been written for the pipe in continuous transfer mode, the USB sets the BVAL flag to 1 and switches the FIFO buffer from the CPU side to the SIE side, thus enabling transmission.
Writing 1 to the BVAL flag should be done while the FRDY bit is 1 (set by the USB).
When the selected pipe is in the receiving direction, do not set the BVAL flag to 1.

## 25.2.7 Interrupt Enable Register 0 (INTENB0)

Address(es): 000A 0030h

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | VBSE | RSME | SOFE | DVSE | CTRE | BEMPE | NRDYE | BRDYE | — | — | — | — | — | — | — | — |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b7 to b0 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |
| b8 | BRDYE | Buffer Ready Interrupt Enable | 0: Interrupt output disabled<br>1: Interrupt output enabled | R/W |
| b9 | NRDYE | Buffer Not Ready Response Interrupt Enable | 0: Interrupt output disabled<br>1: Interrupt output enabled | R/W |
| b10 | BEMPE | Buffer Empty Interrupt Enable | 0: Interrupt output disabled<br>1: Interrupt output enabled | R/W |
| b11 | CTRE | Control Transfer Stage Transition Interrupt Enable*1 | 0: Interrupt output disabled<br>1: Interrupt output enabled | R/W |
| b12 | DVSE | Device State Transition Interrupt Enable*1 | 0: Interrupt output disabled<br>1: Interrupt output enabled | R/W |
| b13 | SOFE | Frame Number Update Interrupt Enable | 0: Interrupt output disabled<br>1: Interrupt output enabled | R/W |
| b14 | RSME | Resume Interrupt Enable*1 | 0: Interrupt output disabled<br>1: Interrupt output enabled | R/W |
| b15 | VBSE | VBUS Interrupt Enable | 0: Interrupt output disabled<br>1: Interrupt output enabled | R/W |

Note 1. The RSME, DVSE, and CTRE bits can be set to 1 only when the function controller is selected; do not set these bits to 1 to enable the corresponding interrupt output when the host controller is selected.

On detecting the interrupt corresponding to the bit in INTENB0 to which 1 has been set by software, the USB generates the USB interrupt request.

The USB sets 1 to each status bit in INTSTS0 when a detection condition of the corresponding interrupt source has been satisfied regardless of the setting in INTENB0 (regardless of whether the interrupt output is enabled or disabled).

While the status bit in INTSTS0 corresponding to the interrupt source indicates 1, the USB generates the USB interrupt request when the corresponding interrupt enable bit in INTENB0 is modified from 0 to 1 by software.

## 25.2.8 Interrupt Enable Register 1 (INTENB1)

Address(es): 000A 0032h

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | OVRCRE | BCHGE | — | DTCHE | ATTCHE | — | — | — | — | EOFERRE | SIGNE | SACKE | — | — | — | PDDETINTE0 |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b0 | PDDETINTE0 | PDDETINT0 Detection Interrupt Enable | 0: Interrupt output disabled<br>1: Interrupt output enabled | R/W |
| b3 to b1 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |
| b4 | SACKE | Setup Transaction Normal Response Interrupt Enable | 0: Interrupt output disabled<br>1: Interrupt output enabled | R/W |
| b5 | SIGNE | Setup Transaction Error Interrupt Enable | 0: Interrupt output disabled<br>1: Interrupt output enabled | R/W |
| b6 | EOFERRE | EOF Error Detection Interrupt Enable | 0: Interrupt output disabled<br>1: Interrupt output enabled | R/W |
| b10 to b7 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |
| b11 | ATTCHE | Connection Detection Interrupt Enable | 0: Interrupt output disabled<br>1: Interrupt output enabled | R/W |
| b12 | DTCHE | Disconnection Detection Interrupt Enable | 0: Interrupt output disabled<br>1: Interrupt output enabled | R/W |
| b13 | — | Reserved | This bit is read as 0. The write value should be 0. | R/W |
| b14 | BCHGE | USB Bus Change Interrupt Enable | 0: Interrupt output disabled<br>1: Interrupt output enabled | R/W |
| b15 | OVRCRE | Overcurrent Input Change Interrupt Enable | 0: Interrupt output disabled<br>1: Interrupt output enabled | R/W |

Note: • The bits in INTENB1 can be set to 1 only when the host controller is selected; do not set these bits to 1 to enable the corresponding interrupt output when the function controller is selected.

INTENB1 specifies the interrupt masks when the host controller is selected, and for the setup transaction.
On detecting the interrupt corresponding to the bit in INTENB1 to which 1 has been set by software, the USB generates the USB interrupt request.
The USB sets 1 to each status bit in INTSTS1 when a detection condition of the corresponding interrupt source has been satisfied regardless of the setting in INTENB1 (regardless of whether the interrupt output is enabled or disabled).
While the status bit in INTSTS1 corresponding to the interrupt source indicates 1, the USB generates the USB interrupt request when the corresponding interrupt enable bit in INTENB1 is modified from 0 to 1 by software.
When the function controller is selected, the interrupts should not be enabled.

## 25.2.9 BRDY Interrupt Enable Register (BRDYENB)

Address(es): 000A 0036h

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | PIPE9BRDYE | PIPE8BRDYE | PIPE7BRDYE | PIPE6BRDYE | PIPE5BRDYE | PIPE4BRDYE | PIPE3BRDYE | PIPE2BRDYE | PIPE1BRDYE | PIPE0BRDYE |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b0 | PIPE0BRDYE | BRDY Interrupt Enable for PIPE0 | 0: Interrupt output disabled<br>1: Interrupt output enabled | R/W |
| b1 | PIPE1BRDYE | BRDY Interrupt Enable for PIPE1 | 0: Interrupt output disabled<br>1: Interrupt output enabled | R/W |
| b2 | PIPE2BRDYE | BRDY Interrupt Enable for PIPE2 | 0: Interrupt output disabled<br>1: Interrupt output enabled | R/W |
| b3 | PIPE3BRDYE | BRDY Interrupt Enable for PIPE3 | 0: Interrupt output disabled<br>1: Interrupt output enabled | R/W |
| b4 | PIPE4BRDYE | BRDY Interrupt Enable for PIPE4 | 0: Interrupt output disabled<br>1: Interrupt output enabled | R/W |
| b5 | PIPE5BRDYE | BRDY Interrupt Enable for PIPE5 | 0: Interrupt output disabled<br>1: Interrupt output enabled | R/W |
| b6 | PIPE6BRDYE | BRDY Interrupt Enable for PIPE6 | 0: Interrupt output disabled<br>1: Interrupt output enabled | R/W |
| b7 | PIPE7BRDYE | BRDY Interrupt Enable for PIPE7 | 0: Interrupt output disabled<br>1: Interrupt output enabled | R/W |
| b8 | PIPE8BRDYE | BRDY Interrupt Enable for PIPE8 | 0: Interrupt output disabled<br>1: Interrupt output enabled | R/W |
| b9 | PIPE9BRDYE | BRDY Interrupt Enable for PIPE9 | 0: Interrupt output disabled<br>1: Interrupt output enabled | R/W |
| b15 to b10 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |

BRDYENB enables or disables the INTSTS0.BRDY bit to be set to 1 when the BRDY interrupt is detected for each pipe.

On detecting the BRDY interrupt for the pipe corresponding to the bit in BRDYENB to which 1 has been set by software, the USB sets 1 to the corresponding PIPEnBRDY bit (n = 0 to 9) and the INTSTS0.BRDY bit, and generates the BRDY interrupt request.

While at least one PIPEnBRDY bit indicates 1, the USB generates the BRDY interrupt request when the corresponding interrupt enable bit in BRDYENB is modified from 0 to 1 by software.

## 25.2.10 NRDY Interrupt Enable Register (NRDYENB)

Address(es): 000A 0038h

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | PIPE9NRDYE | PIPE8NRDYE | PIPE7NRDYE | PIPE6NRDYE | PIPE5NRDYE | PIPE4NRDYE | PIPE3NRDYE | PIPE2NRDYE | PIPE1NRDYE | PIPE0NRDYE |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b0 | PIPE0NRDYE | NRDY Interrupt Enable for PIPE0 | 0: Interrupt output disabled<br>1: Interrupt output enabled | R/W |
| b1 | PIPE1NRDYE | NRDY Interrupt Enable for PIPE1 | 0: Interrupt output disabled<br>1: Interrupt output enabled | R/W |
| b2 | PIPE2NRDYE | NRDY Interrupt Enable for PIPE2 | 0: Interrupt output disabled<br>1: Interrupt output enabled | R/W |
| b3 | PIPE3NRDYE | NRDY Interrupt Enable for PIPE3 | 0: Interrupt output disabled<br>1: Interrupt output enabled | R/W |
| b4 | PIPE4NRDYE | NRDY Interrupt Enable for PIPE4 | 0: Interrupt output disabled<br>1: Interrupt output enabled | R/W |
| b5 | PIPE5NRDYE | NRDY Interrupt Enable for PIPE5 | 0: Interrupt output disabled<br>1: Interrupt output enabled | R/W |
| b6 | PIPE6NRDYE | NRDY Interrupt Enable for PIPE6 | 0: Interrupt output disabled<br>1: Interrupt output enabled | R/W |
| b7 | PIPE7NRDYE | NRDY Interrupt Enable for PIPE7 | 0: Interrupt output disabled<br>1: Interrupt output enabled | R/W |
| b8 | PIPE8NRDYE | NRDY Interrupt Enable for PIPE8 | 0: Interrupt output disabled<br>1: Interrupt output enabled | R/W |
| b9 | PIPE9NRDYE | NRDY Interrupt Enable for PIPE9 | 0: Interrupt output disabled<br>1: Interrupt output enabled | R/W |
| b15 to b10 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |

NRDYENB enables or disables the INTSTS0.NRDY bit to be set to 1 when the NRDY interrupt is detected for each pipe.

On detecting the NRDY interrupt for the pipe corresponding to the bit in NRDYENB to which 1 has been set by software, the USB sets 1 to the corresponding PIPEnNRDY bit (n = 0 to 9) and the INTSTS0.NRDY bit, and generates the NRDY interrupt request.

While at least one PIPEnNRDY bit indicates 1, the USB generates the NRDY interrupt request when the corresponding interrupt enable bit in NRDYENB is modified from 0 to 1 by software.

## 25.2.11 BEMP Interrupt Enable Register (BEMPENB)

Address(es): 000A 003Ah

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | PIPE9BEMPE | PIPE8BEMPE | PIPE7BEMPE | PIPE6BEMPE | PIPE5BEMPE | PIPE4BEMPE | PIPE3BEMPE | PIPE2BEMPE | PIPE1BEMPE | PIPE0BEMPE |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b0 | PIPE0BEMPE | BEMP Interrupt Enable for PIPE0 | 0: Interrupt output disabled<br>1: Interrupt output enabled | R/W |
| b1 | PIPE1BEMPE | BEMP Interrupt Enable for PIPE1 | 0: Interrupt output disabled<br>1: Interrupt output enabled | R/W |
| b2 | PIPE2BEMPE | BEMP Interrupt Enable for PIPE2 | 0: Interrupt output disabled<br>1: Interrupt output enabled | R/W |
| b3 | PIPE3BEMPE | BEMP Interrupt Enable for PIPE3 | 0: Interrupt output disabled<br>1: Interrupt output enabled | R/W |
| b4 | PIPE4BEMPE | BEMP Interrupt Enable for PIPE4 | 0: Interrupt output disabled<br>1: Interrupt output enabled | R/W |
| b5 | PIPE5BEMPE | BEMP Interrupt Enable for PIPE5 | 0: Interrupt output disabled<br>1: Interrupt output enabled | R/W |
| b6 | PIPE6BEMPE | BEMP Interrupt Enable for PIPE6 | 0: Interrupt output disabled<br>1: Interrupt output enabled | R/W |
| b7 | PIPE7BEMPE | BEMP Interrupt Enable for PIPE7 | 0: Interrupt output disabled<br>1: Interrupt output enabled | R/W |
| b8 | PIPE8BEMPE | BEMP Interrupt Enable for PIPE8 | 0: Interrupt output disabled<br>1: Interrupt output enabled | R/W |
| b9 | PIPE9BEMPE | BEMP Interrupt Enable for PIPE9 | 0: Interrupt output disabled<br>1: Interrupt output enabled | R/W |
| b15 to b10 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |

BEMPENB enables or disables the INTSTS0.BEMP bit to be set to 1 when the BEMP interrupt is detected for each pipe. On detecting the BEMP interrupt for the pipe corresponding to the bit in BEMPENB to which 1 has been set by software, the USB sets 1 to the corresponding PIPEnBEMP bit (n = 0 to 9) and the INTSTS0.BEMP bit, and generates the BEMP interrupt request.

While at least one PIPEnBEMP bit in BEMPSTS indicates 1, the USB generates the BEMP interrupt request when the corresponding interrupt enable bit in BEMPENB is modified from 0 to 1 by software.

## 25.2.12 SOF Output Configuration Register (SOFCFG)

Address(es): 000A 003Ch

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | TRNENSEL | — | BRDYM | — | EDGESTS | — | — | — | — |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b3 to b0 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |
| b4 | EDGESTS | Edge Interrupt Output Status Monitor *1 | Indicates 1 when the edge interrupt output signal is in the middle of the edge processing. | R |
| b5 | — | Reserved | This bit is read as 0. The write value should be 0. | R/W |
| b6 | BRDYM | BRDY Interrupt Status Clear Timing | 0: Software clears the status.<br>1: The USB clears the status when data has been read from the FIFO buffer or data has been written to the FIFO buffer. | R/W |
| b7 | — | Reserved | This bit is read as 0. The write value should be 0. | R/W |
| b8 | TRNENSEL | Transaction-Enabled Time Select *1 | 0: For non-low-speed communication<br>1: For low-speed communication | R/W |
| b15 to b9 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |

Note 1. Make sure that this bit is 0 when stopping the clock supply to the USB module.

### EDGESTS Bit (Edge Interrupt Output Status Monitor)

The EDGESTS bit indicates 1 when the edge interrupt output signal is in the middle of the edge processing.
Make sure the EDGESTS bit is 0 when stopping the clock supply to the USB.

### BRDYM Bit (BRDY Interrupt Status Clear Timing)

The BRDYM bit specifies the timing for clearing the BRDY interrupt status for each pipe.

### TRNENSEL Bit (Transaction-Enabled Time Select)

The TRNENSEL bit selects, for full-speed or low-speed communication, the transaction-enabled time in which the USB issues tokens in a frame via the port.
Set the TRNENSEL bit to 1 when a low-speed device is connected via the HUB.
The TRNENSEL bit is valid only when the host controller is selected.
This bit should be set to 0 if the function controller is selected.

## 25.2.13 Interrupt Status Register 0 (INTSTS0)

Address(es): 000A 0040h

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | VBINT | RESM | SOFR | DVST | CTRT | BEMP | NRDY | BRDY | VBSTS | DVSQ[2:0] | | | VALID | CTSQ[2:0] | | |
| Value after reset: | 0 | 0 | 0 | 0/1 *1 | 0 | 0 | 0 | 0 | 0 *2 | 0 | 0 | 0/1 *3 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b2 to b0 | CTSQ[2:0] | Control Transfer Stage | b2 b0<br>0 0 0: Idle or setup stage<br>0 0 1: Control read data stage<br>0 1 0: Control read status stage<br>0 1 1: Control write data stage<br>1 0 0: Control write status stage<br>1 0 1: Control write (no data) status stage<br>1 1 0: Control transfer sequence error | R |
| b3 | VALID | USB Request Reception | 0: Setup packet is not received<br>1: Setup packet is received | R/W |
| b6 to b4 | DVSQ[2:0] | Device State | b6 b4<br>0 0 0: Powered state<br>0 0 1: Default state<br>0 1 0: Address state<br>0 1 1: Configured state<br>1 x x: Suspended state | R |
| b7 | VBSTS | VBUS Input Status | 0: USB0_VBUS pin is low.<br>1: USB0_VBUS pin is high. | R |
| b8 | BRDY | Buffer Ready Interrupt Status | 0: BRDY interrupts are not generated.<br>1: BRDY interrupts are generated. | R |
| b9 | NRDY | Buffer Not Ready Interrupt Status | 0: NRDY interrupts are not generated.<br>1: NRDY interrupts are generated. | R |
| b10 | BEMP | Buffer Empty Interrupt Status | 0: BEMP interrupts are not generated.<br>1: BEMP interrupts are generated. | R |
| b11 | CTRT | Control Transfer Stage Transition Interrupt Status*5 | 0: Control transfer stage transition interrupts are not generated.<br>1: Control transfer stage transition interrupts are generated. | R/W *4 |
| b12 | DVST | Device State Transition Interrupt Status*5 | 0: Device state transition interrupts are not generated.<br>1: Device state transition interrupts are generated. | R/W *4 |
| b13 | SOFR | Frame Number Refresh Interrupt Status | 0: SOF interrupts are not generated.<br>1: SOF interrupts are generated. | R/W *4 |
| b14 | RESM | Resume Interrupt Status*5,*6 | 0: Resume interrupts are not generated.<br>1: Resume interrupts are generated. | R/W *4 |
| b15 | VBINT | VBUS Interrupt Status*6 | 0: VBUS interrupts are not generated.<br>1: VBUS interrupts are generated. | R/W *4 |

Note 1. The value is 0 after a power-on reset and 1 after a USB bus reset.
Note 2. The value is 1 when the USB0_VBUS pin is high and 0 when the USB0_VBUS pin is low.
Note 3. The value is 000b after a power-on reset and 001b after a USB bus reset.
Note 4. To clear the VBINT, RESM, SOFR, DVST, CTRT, or VALID bit, write 0 only to the bits to be cleared; write 1 to the other bits. Do not write 0 to the status bits indicating 0.
Note 5. The status of the RESM, DVST, and CTRT bits are changed only when the host controller is selected. Set the corresponding interrupt enable bits to 0 (disabled) when the function controller is selected.
Note 6. A change in the status indicated by the VBINT and RESM bits can be detected even while the clock supply is stopped (the SCKE bit = 0), and the interrupts are output when the corresponding interrupt enable bits are enabled. Clearing the status through software should be done after enabling the clock supply.

**CTSQ[2:0] Bits (Control Transfer Stage)**

When the host controller is selected, the read value is invalid.

**VALID Bit (USB Request Reception)**

When the host controller is selected, the read value is invalid.

**DVSQ[2:0] Bits (Device State)**

The DVSQ[2:0] bits are initialized by a USB bus reset.
When the host controller is selected, the read value is invalid.

**BRDY Bit (Buffer Ready Interrupt Status)**

Indicates the BRDY interrupt status.
The USB sets the BRDY bit to 1 when at least one PIPEnBRDY bit (n = 0 to 9) is set to 1 among the PIPEBRDY bits. These bits correspond to the BRDYENB.PIPEnBRDYE bits (n = 0 to 9) to which 1 has been set, when the USB detects the BRDY interrupt status in at least one pipe among the pipes for which the BRDY interrupt output is enabled by software.
For the conditions for PIPEnBRDY status assertion, refer to section 25.3.3.1, BRDY Interrupt.
The USB clears the BRDY bit to 0 when all the PIPEnBRDY bits corresponding to the PIPEnBRDYE bits to which 1 has been set are written to 0 by software.
The BRDY bit cannot be cleared to 0 even if this bit is written to 0 by software.

**NRDY Bit (Buffer Not Ready Interrupt Status)**

The USB sets the NRDY bit to 1 when at least one PIPEnNRDY bit (n = 0 to 9) is set to 1 among the PIPENRDY bits corresponding to the PIPEnNRDYE bits (n = 0 to 9) to which 1 has been set (when the USB detects the NRDY interrupt status in at least one pipe among the pipes for which software enables the NRDY interrupt output).
For the conditions for PIPEnNRDY status assertion, refer to section 25.3.3.2, NRDY Interrupt.
The USB clears the NRDY bit to 0 when all the PIPEnNRDY bits corresponding to the PIPEnNRDYE bits to which 1 has been set are written to 0 by software.
The NRDY bit cannot be cleared to 0 even if this bit is written to 0 by software.

**BEMP Bit (Buffer Empty Interrupt Status)**

The USB sets the BEMP bit to 1 when at least one PIPEnBEMP bit (n = 0 to 9) is set to 1 among the PIPEnBEMP bits corresponding to the PIPEnBEMPE bits (n = 0 to 9) to which 1 has been set (when the USB detects the BEMP interrupt status in at least one pipe among the pipes for which the BEMP interrupt output is enabled by software).
For the conditions for PIPEnBEMP status assertion, refer to section 25.3.3.3, BEMP Interrupt.
The USB clears the BEMP bit to 0 when all the PIPEnBEMP bits corresponding to the PIPEnBEMPE bits to which 1 has been set are written to 0 by software.
The BEMP bit cannot be cleared to 0 even if this bit is written to 0 by software.

**CTRT Bit (Control Transfer Stage Transition Interrupt Status)**

When the function controller is selected, the USB updates the value of the CTSQ[2:0] bits and sets the CTRT bit to 1 on detecting a change in the control transfer stage.
When a control transfer stage transition interrupt is generated, clear the status before the USB detects the next control transfer stage transition.
When the host controller is selected, the read value is invalid.

#### DVST Bit (Device State Transition Interrupt Status)

When the function controller is selected, the USB updates the DVSQ[2:0] value and sets the DVST bit to 1 on detecting a change in the device state.
When a device state transition interrupt is generated, clear the status before the USB detects the next device state transition.
When the host controller is selected, the read value is invalid.

#### SOFR Bit (Frame Number Refresh Interrupt Status)

(1) When the host controller is selected
The USB sets the SOFR bit to 1 on updating the frame number when the DVSTCTR0.UACT bit has been set to 1 by software. (A frame number refresh interrupt is detected every 1 ms.)
(2) When the function controller is selected
The USB sets the SOFR bit to 1 on updating the frame number. (A frame number refresh interrupt is detected every 1 ms.)
The USB can detect an SOFR interrupt through the internal interpolation function even when a damaged SOF packet is received from the USB host.

#### RESM Bit (Resume Interrupt Status)

When the function controller is selected, the USB sets the RESM bit to 1 on detecting the falling edge of the signal on the USB0_DP pin in the suspended state (DVSQ[2:0] = 1xxb).
When the host controller is selected, the read value is invalid.

#### VBINT Bit (VBUS Interrupt Status)

The USB sets the VBINT bit to 1 on detecting a level change (high to low or low to high) in the USB0_VBUS pin input value. The USB sets the VBSTS bit to indicate the USB0_VBUS pin input value. When the VBUS interrupt is generated, use software to repeat reading the VBSTS bit until the same value is read three or more times, and eliminate chattering.

## 25.2.14 Interrupt Status Register 1 (INTSTS1)

Address(es): 000A 0042h

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | OVRCR | BCHG | — | DTCH | ATTCH | — | — | — | — | EOFERR | SIGN | SACK | — | — | — | PDDETINT0 |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b0 | PDDETINT0 | PDDET0 Detection Interrupt Status | 0: PDDET0 detection interrupts are not generated.<br>1: PDDET0 detection interrupts are generated. | R/W *1 |
| b3 to b1 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |
| b4 | SACK | Setup Transaction Normal Response Interrupt Status | 0: SACK interrupts are not generated.<br>1: SACK interrupts are generated. | R/W *1 |
| b5 | SIGN | Setup Transaction Error Interrupt Status | 0: SIGN interrupts are not generated.<br>1: SIGN interrupts are generated. | R/W *1 |
| b6 | EOFERR | EOF Error Detection Interrupt Status | 0: EOFERR interrupts are not generated.<br>1: EOFERR interrupts are generated. | R/W *1 |
| b10 to b7 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |
| b11 | ATTCH | ATTCH Interrupt Status | 0: ATTCH interrupts are not generated.<br>1: ATTCH interrupts are generated. | R/W *1 |
| b12 | DTCH | USB Disconnection Detection Interrupt Status | 0: DTCH interrupts are not generated.<br>1: DTCH interrupts are generated. | R/W *1 |
| b13 | — | Reserved | This bit is read as 0. The write value should be 0. | R/W |
| b14 | BCHG | USB Bus Change Interrupt Status*2 | 0: BCHG interrupts are not generated.<br>1: BCHG interrupts are generated. | R/W *1 |
| b15 | OVRCR | Overcurrent Input Change Interrupt Status*2 | 0: OVRCR interrupts are not generated.<br>1: OVRCR interrupts are generated. | R/W *1 |

Note 1. To clear the status indicated by the bits in INTSTS1, write 0 only to the bits to be cleared; write 1 to the other bits.
Note 2. A change in the status indicated by the OVRCR or BCHG bit can be detected even while the clock supply is stopped (while the SYSCFG.SCKE bit = 0), and the interrupt is output when the corresponding interrupt enable bit is enabled. Clearing the status through software should be done after setting the SYSCFG.SCKE bit to 1.
No interrupts other than those indicated by the BCHG and OVRCR bits can be detected while the clock supply is stopped (while the SYSCFG.SCKE bit = 0).

INTSTS1 is used to confirm the status of each interrupt when the host controller is selected.
The various status change interrupts indicated by the bits in INTSTS1 should be enabled only when the host controller is selected.

### PDDETINT Bit (Portable Device Detection Interrupt Status)

Indicates the status of the portable device detection interrupt when the host controller is selected.
This bit is set to 1 when the USB module detects when a level change (high to low or low to high) occurs in the input value to the VDPDET pin of the USB physical layer transceiver (PHY). The USB module sets the PDDETSTS bit to indicate the VDPDET input value.
When the PDDETINT interrupt is generated, use software to repeat reading the PDDETSTS bit until the same value is read three or more times, and eliminate chattering.

#### SACK Bit (Setup Transaction Normal Response Interrupt Status)

Indicates the status of the setup transaction normal response interrupt when the host controller is selected.
The USB detects the SACK interrupt when ACK response is returned from the peripheral device during the setup transactions issued by the USB, and sets the SACK bit to 1. Here, if the corresponding interrupt enable bit has been set to 1 by software, the USB generates the SACK interrupt.
When the function controller is selected, the read value is invalid.

#### SIGN Bit (Setup Transaction Error Interrupt Status)

Indicates the status of the setup transaction error interrupt when the host controller is selected.
The USB detects the SIGN interrupt when ACK response is not returned from the peripheral device three consecutive times during the setup transactions issued by this module, and sets the SIGN bit to 1. Here, if the corresponding interrupt enable bit has been set to 1 by software, the USB generates the SIGN interrupt.
Specifically, the USB detects the SIGN interrupt when any of the following response conditions occur for three consecutive setup transactions.

- Timeout is detected by the USB when the peripheral device has returned no response.
- A damaged ACK packet is received.
- A handshake other than ACK (NAK, NYET, or STALL) is received.

When the function controller is selected, the read value is invalid.

#### EOFERR Bit (EOF Error Detection Interrupt Status)

Indicates the status of the EOFERR interrupt when the host controller is selected.
The USB detects the EOFERR interrupt on detecting that communication is not completed at the EOF2 timing prescribed by USB Specifications 2.0, and sets the EOFERR bit to 1. Here, if the corresponding interrupt enable bit has been set to 1 by software, the USB generates the EOFERR interrupt.
After detecting the EOFERR interrupt, the USB controls hardware as described below (irrespective of the setting of the corresponding interrupt enable bit). All pipes in which communications are currently carried out for the USB port should be terminated by software and perform re-enumeration of the USB port.

- Modifies the DVSTCTR0.UACT bit for the port in which an EOFERR interrupt has been detected to 0.
- Puts the port in which an EOFERR interrupt has been generated into the idle state.

When the function controller is selected, the read value is invalid.

#### ATTCH Bit (ATTCH Interrupt Status)

Indicates the status of the ATTCH interrupt when the host controller is selected.
The USB detects the ATTCH interrupt on detecting J-state or K-state of the full-speed or low-speed signal level for 2.5 µs, and sets the ATTCH bit to 1. Here, if the corresponding interrupt enable bit has been set to 1 by software, the USB generates the interrupt.
Specifically, the USB detects the ATTCH interrupt on any of the following conditions.

- K-state, SE0, or SE1 changes to J-state, and J-state continues for 2.5 µs.
- J-state, SE0, or SE1 changes to K-state, and K-state continues for 2.5 µs.

When the function controller is selected, the read value is invalid.

### DTCH Bit (USB Disconnection Detection Interrupt Status)

Indicates the status of the USB disconnection detection interrupt when the host controller is selected.
The USB detects the DTCH interrupt on detecting USB bus disconnection, and sets the DTCH bit to 1. Here, if the corresponding interrupt enable bit has been set to 1 by software, the USB generates the interrupt.
The USB detects bus disconnection based on USB Specifications 2.0.
After detecting the DTCH interrupt, the USB controls hardware as described below (irrespective of the setting of the corresponding interrupt enable bit). All the pipes in which communications are currently carried out for the USB port should be terminated by software and make a transition to the wait state for bus connection to the USB port (wait state for ATTCH interrupt generation).

- Modifies the DVSTCTR0.UACT bit for the port in which a DTCH interrupt has been detected to 0.
- Puts the port in which a DTCH interrupt has been generated into the idle state.

When the function controller is selected, the read value is invalid.

### BCHG Bit (USB Bus Change Interrupt Status)

Indicates the status of the USB bus change interrupt.
The USB detects the BCHG interrupt when a change in the full-speed or low-speed signal level occurs on the USB port (a change from J-state, K-state, or SE0 to J-state, K-state, or SE0), and sets the BCHG bit to 1. Here, if the corresponding interrupt enable bit has been set to 1 by software, the USB generates the interrupt.
The USB sets the LNST[1:0] bits to indicate the current input state of the USB port. When the BCHG interrupt is generated, use software to repeat reading the LNST[1:0] bits until the same value is read three or more times, and eliminate chattering.
A change in the USB bus state can be detected even while the internal clock supply is stopped.
When the function controller is selected, the read value is invalid.

### OVRCR Bit (Overcurrent Input Change Interrupt Status)

Indicates the status of the USB0_OVRCURA and USB0_OVRCURB input pin change interrupt.
The USB detects the OVRCR interrupt when a change (high to low or low to high) occurs in at least one of the input values to the USB0_OVRCURA and USB0_OVRCURB pins, and sets the OVRCR bit to 1. Here, if the corresponding interrupt enable bit has been set to 1 by software, the USB generates the interrupt.

## 25.2.15 BRDY Interrupt Status Register (BRDYSTS)

Address(es): 000A 0046h

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | PIPE9BRDY | PIPE8BRDY | PIPE7BRDY | PIPE6BRDY | PIPE5BRDY | PIPE4BRDY | PIPE3BRDY | PIPE2BRDY | PIPE1BRDY | PIPE0BRDY |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b0 | PIPE0BRDY | BRDY Interrupt Status for PIPE0*2 | 0: Interrupts are not generated.<br>1: Interrupts are generated. | R/W *1 |
| b1 | PIPE1BRDY | BRDY Interrupt Status for PIPE1*2 | 0: Interrupts are not generated.<br>1: Interrupts are generated. | R/W *1 |
| b2 | PIPE2BRDY | BRDY Interrupt Status for PIPE2*2 | 0: Interrupts are not generated.<br>1: Interrupts are generated. | R/W *1 |
| b3 | PIPE3BRDY | BRDY Interrupt Status for PIPE3*2 | 0: Interrupts are not generated.<br>1: Interrupts are generated. | R/W *1 |
| b4 | PIPE4BRDY | BRDY Interrupt Status for PIPE4*2 | 0: Interrupts are not generated.<br>1: Interrupts are generated. | R/W *1 |
| b5 | PIPE5BRDY | BRDY Interrupt Status for PIPE5*2 | 0: Interrupts are not generated.<br>1: Interrupts are generated. | R/W *1 |
| b6 | PIPE6BRDY | BRDY Interrupt Status for PIPE6*2 | 0: Interrupts are not generated.<br>1: Interrupts are generated. | R/W *1 |
| b7 | PIPE7BRDY | BRDY Interrupt Status for PIPE7*2 | 0: Interrupts are not generated.<br>1: Interrupts are generated. | R/W *1 |
| b8 | PIPE8BRDY | BRDY Interrupt Status for PIPE8*2 | 0: Interrupts are not generated.<br>1: Interrupts are generated. | R/W *1 |
| b9 | PIPE9BRDY | BRDY Interrupt Status for PIPE9*2 | 0: Interrupts are not generated.<br>1: Interrupts are generated. | R/W *1 |
| b15 to b10 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |

Note 1. When the SOFCFG.BRDYM bit is set to 0, to clear the status indicated by the bits in BRDYSTS, write 0 only to the bits to be cleared; write 1 to the other bits.

Note 2. When the SOFCFG.BRDYM bit is set to 0, clearing BRDY Interrupts should be done before accessing the FIFO.

## 25.2.16 NRDY Interrupt Status Register (NRDYSTS)

Address(es): 000A 0048h

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | PIPE9NRDY | PIPE8NRDY | PIPE7NRDY | PIPE6NRDY | PIPE5NRDY | PIPE4NRDY | PIPE3NRDY | PIPE2NRDY | PIPE1NRDY | PIPE0NRDY |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b0 | PIPE0NRDY | NRDY Interrupt Status for PIPE0 | 0: Interrupts are not generated.<br>1: Interrupts are generated. | R/W *1 |
| b1 | PIPE1NRDY | NRDY Interrupt Status for PIPE1 | 0: Interrupts are not generated.<br>1: Interrupts are generated. | R/W *1 |
| b2 | PIPE2NRDY | NRDY Interrupt Status for PIPE2 | 0: Interrupts are not generated.<br>1: Interrupts are generated. | R/W *1 |
| b3 | PIPE3NRDY | NRDY Interrupt Status for PIPE3 | 0: Interrupts are not generated.<br>1: Interrupts are generated. | R/W *1 |
| b4 | PIPE4NRDY | NRDY Interrupt Status for PIPE4 | 0: Interrupts are not generated.<br>1: Interrupts are generated. | R/W *1 |
| b5 | PIPE5NRDY | NRDY Interrupt Status for PIPE5 | 0: Interrupts are not generated.<br>1: Interrupts are generated. | R/W *1 |
| b6 | PIPE6NRDY | NRDY Interrupt Status for PIPE6 | 0: Interrupts are not generated.<br>1: Interrupts are generated. | R/W *1 |
| b7 | PIPE7NRDY | NRDY Interrupt Status for PIPE7 | 0: Interrupts are not generated.<br>1: Interrupts are generated. | R/W *1 |
| b8 | PIPE8NRDY | NRDY Interrupt Status for PIPE8 | 0: Interrupts are not generated.<br>1: Interrupts are generated. | R/W *1 |
| b9 | PIPE9NRDY | NRDY Interrupt Status for PIPE9 | 0: Interrupts are not generated.<br>1: Interrupts are generated. | R/W *1 |
| b15 to b10 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |

Note 1. To clear the status indicated by the bits in NRDYSTS, write 0 only to the bits to be cleared; write 1 to the other bits.

## 25.2.17 BEMP Interrupt Status Register (BEMPSTS)

Address(es): 000A 004Ah

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | PIPE9BEMP | PIPE8BEMP | PIPE7BEMP | PIPE6BEMP | PIPE5BEMP | PIPE4BEMP | PIPE3BEMP | PIPE2BEMP | PIPE1BEMP | PIPE0BEMP |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b0 | PIPE0BEMP | BEMP Interrupt Status for PIPE0 | 0: Interrupts are not generated.<br>1: Interrupts are generated. | R/W *1 |
| b1 | PIPE1BEMP | BEMP Interrupt Status for PIPE1 | 0: Interrupts are not generated.<br>1: Interrupts are generated. | R/W *1 |
| b2 | PIPE2BEMP | BEMP Interrupt Status for PIPE2 | 0: Interrupts are not generated.<br>1: Interrupts are generated. | R/W *1 |
| b3 | PIPE3BEMP | BEMP Interrupt Status for PIPE3 | 0: Interrupts are not generated.<br>1: Interrupts are generated. | R/W *1 |
| b4 | PIPE4BEMP | BEMP Interrupt Status for PIPE4 | 0: Interrupts are not generated.<br>1: Interrupts are generated. | R/W *1 |
| b5 | PIPE5BEMP | BEMP Interrupt Status for PIPE5 | 0: Interrupts are not generated.<br>1: Interrupts are generated. | R/W *1 |
| b6 | PIPE6BEMP | BEMP Interrupt Status for PIPE6 | 0: Interrupts are not generated.<br>1: Interrupts are generated. | R/W *1 |
| b7 | PIPE7BEMP | BEMP Interrupt Status for PIPE7 | 0: Interrupts are not generated.<br>1: Interrupts are generated. | R/W *1 |
| b8 | PIPE8BEMP | BEMP Interrupt Status for PIPE8 | 0: Interrupts are not generated.<br>1: Interrupts are generated. | R/W *1 |
| b9 | PIPE9BEMP | BEMP Interrupt Status for PIPE9 | 0: Interrupts are not generated.<br>1: Interrupts are generated. | R/W *1 |
| b15 to b10 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |

Note 1. To clear the status indicated by the bits in BEMPSTS, write 0 only to the bits to be cleared; write 1 to the other bits.

## 25.2.18 Frame Number Register (FRMNUM)

Address(es): 000A 004Ch

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | OVRN | CRCE | — | — | — | FRNM[10:0] | | | | | | | | | | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b10 to b0 | FRNM[10:0] | Frame Number | Latest frame number | R |
| b13 to b11 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |
| b14 | CRCE | Receive Data Error | 0: No error<br>1: An error occurred | R/W<br>*1 |
| b15 | OVRN | Overrun/Underrun Detection Status | 0: No error<br>1: An error occurred | R/W<br>*1 |

Note 1. Only 0 can be written.

### FRNM[10:0] Bits (Frame Number)

These bits indicate the latest frame number for the USB after the issuing of an SOF packet every 1 ms or writing to the FRNM[10:0] bits at the SOF packet reception.
The CRCE bit can be cleared to 0 by writing 0 to the CRCE bit by software.

### CRCE Bit (Receive Data Error)

Indicates whether a CRC error or bit stuffing error has been detected in the pipe during isochronous transfer.
The CRCE bit can be cleared to 0 by writing 0 to the CRCE bit by software. Here, 1 should be written to the other bits in FRMNUM.
(1) When the host controller is selected
On detecting a CRC error, the USB generates the internal NRDY interrupt request.

### OVRN Bit (Overrun/Underrun Detection Status)

Indicates whether an overrun/underrun error has been detected in the pipe during isochronous transfer.
Software can clear the OVRN bit to 0 by writing 0 to the OVRN bit. Here, 1 should be written to the other bits in FRMNUM.
(1) When the host controller is selected
The USB sets the OVRN bit to 1 on any of the following conditions.

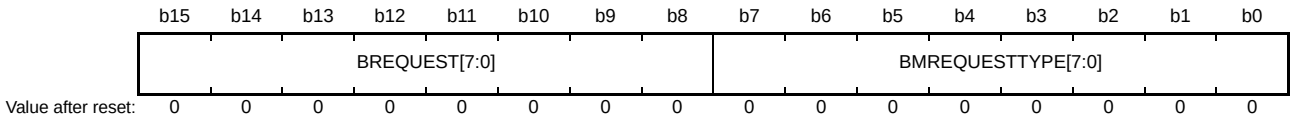
- For the isochronous transfer pipe in the transmitting direction, the time to issue an OUT token comes before all the transmit data has been written to the FIFO buffer.
- For the isochronous transfer pipe in the receiving direction, the time to issue an IN token comes when no FIFO buffer planes are empty.

(2) When the function controller is selected
The USB sets the OVRN bit to 1 on any of the following conditions.

- For the isochronous transfer pipe in the transmitting direction, the IN token is received before all the transmit data has been written to the FIFO buffer.
- For the isochronous transfer pipe in the receiving direction, the OUT token is received when no FIFO buffer planes are empty.

## 25.2.19 USB Request Type Register (USBREQ)

Address(es): 000A 0054h




| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b7 to b0 | BMREQUESTTYPE[7:0] | Request Type | These bits store the USB request bmRequestType value. | R/W *1 |
| b15 tob8 | BREQUEST[7:0] | Request | These bits store the USB request bRequest value. | R/W *1 |

Note 1. When the function controller is selected, these bits can only be read from, and writing to these bits is invalid. When the host controller is selected, these bits can be read from and written to.

USBREQ stores setup requests for control transfers.
When the function controller is selected, the values of bRequest and bmRequestType that have been received are stored.
When the host controller is selected, the values of bRequest and bmRequestType to be transmitted are set.
USBREQ is initialized by a USB bus reset.

### BMREQUESTTYPE[7:0] Bits (Request Type)

These bits hold the value of the bmRequestType field of a USB request.

- When host controller operation is selected:
  Set these bits to the value of the USB request data in setup transactions for transmission. Do not overwrite the value of the BMREQUESTTYPE[7:0] bits while the DCPCTR.SUREQ bit = 1.
- When function controller operation is selected:
  These bits indicate the value of the USB request data in setup transactions for reception. Writing to the bits has no effect.

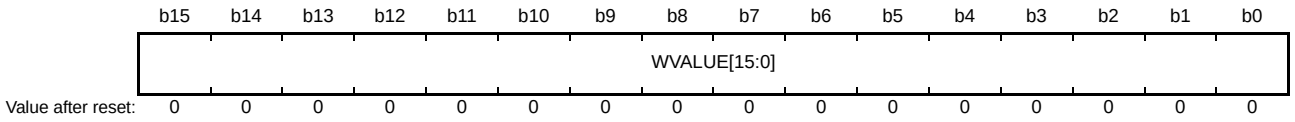
### BREQUEST[7:0] Bits (Request)

These bits store bRequest value of the USB request.

- When host controller operation is selected:
  Set these bits to the value of the USB request data in setup transactions for transmission. Do not overwrite the value of the BREQUEST[7:0] bits while the DCPCTR.SUREQ bit = 1.
- When function controller operation is selected:
  These bits indicate the value of the USB request data in setup transactions for reception. Writing to the bits has no effect.

## 25.2.20 USB Request Value Register (USBVAL)

Address(es): 000A 0056h




| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b15 to b0 | WVALUE[15:0] | Value | These bits store the USB request wValue value. | R/W *1 |

Note 1. When the function controller is selected, these bits can only be read from, and writing to these bits is invalid. When the host controller is selected, these bits can be read from and written to.

When the function controller is selected, the value of wValue that has been received is stored in USBVAL. When the host controller is selected, the value of wValue to be transmitted is set.
USBVAL is initialized by a USB bus reset.

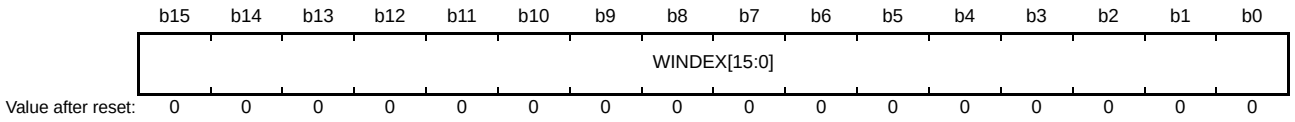
**WVALUE[15:0] Bits (Value)**

These bits store wRequest value of the USB request.

- When host controller operation is selected:
  Set these bits to the value of the wValue field in USB requests of setup transactions for transmission. Do not overwrite the value of the WVALUE[15:0] bits while the DCPCTR.SUREQ bit = 1.
- When function controller operation is selected:
  These bits indicate the value of the wValue field in USB requests received in setup transactions for reception. Writing to the WVALUE[15:0] bits has no effect.

## 25.2.21 USB Request Index Register (USBINDX)

Address(es): 000A 0058h




| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b15 to b0 | WINDEX[15:0] | Index | These bits store the USB request wIndex value. | R/W *1 |

Note 1. When the function controller is selected, these bits can only be read from, and writing to these bits is invalid. When the host controller is selected, these bits can be read from and written to.

USBINDX stores setup requests for control transfers.
When the function controller is selected, the value of wIndex that has been received is stored. When the host controller is selected, the value of wIndex to be transmitted is set.
USBINDX is initialized by a USB bus reset.

### WINDEX[15:0] Bits (Index)

These bits hold the value of the wIndex field of a USB request.

- When host controller operation is selected:
  Set these bits to the value of the wIndex field in USB requests of setup transactions for transmission. Do not overwrite the value of the WINDEX[15:0] bits while the DCPTR.SUREQ bit = 1.
- When function controller operation is selected:
  These bits indicate the value of the wIndex field in USB requests received in setup transactions for reception. Writing to the WINDEX[15:0] bits has no effect.

## 25.2.22 USB Request Length Register (USBLENG)

Address(es): 000A 005Ah

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | WLENGTH[15:0] | | | | | | | | | | | | | | | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b15 to b0 | WLENGTH[15:0] | Length | These bits store the USB request wLength value. | R/W *1 |

Note 1. When the function controller is selected, these bits can only be read from, and writing to these bits is invalid. When the host controller is selected, these bits can be read from and written to.

USBLENG stores setup requests for control transfers.
When the function controller is selected, the value of wLength that has been received is stored. When the host controller is selected, the value of wLength to be transmitted is set.
USBLENG is initialized by a USB bus reset.

### WLENGTH[15:0] Bits (Length)

These bits hold the value of the wLength field of a USB request.

- When host controller operation is selected:
  Set these bits to the value of the wLength field in USB requests of setup transactions for transmission. Do not overwrite the value of the WLENGTH[15:0] bits while the DCPCTR.SUREQ bit = 1.
- When function controller operation is selected:
  These bits indicate the value of the wLength field in USB requests received in setup transactions for reception. Writing to the WLENGTH[15:0] bits has no effect.

## 25.2.23 DCP Configuration Register (DCPCFG)

Address(es): 000A 005Ch

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | SHTNAK | — | — | DIR | — | — | — | — |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b3 to b0 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |
| b4 | DIR | Transfer Direction*1 | 0: Data receiving direction<br>1: Data transmitting direction | R/W |
| b6, b5 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |
| b7 | SHTNAK | Pipe Disabled at End of Transfer*1 | 0: Pipe continued at the end of transfer<br>1: Pipe disabled at the end of transfer | R/W |
| b15 to b8 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |

Note 1. Modify this bit while PID is NAK. Before modifying this bit after modifying the DCPCTR.PID[1:0] bits for the DCP from BUF to NAK, check that the DCPCTR.PBUSY bit is 0. However, if the PID[1:0] bits have been modified to NAK by the USB, checking the PBUSY bit through software is not necessary.

### DIR Bit (Transfer Direction)

When the host controller is selected, the DIR bit sets the transfer direction of the data stage and status stage.
When the function controller is selected, the DIR bit should be set to 0.

### SHTNAK Bit (Pipe Disabled at End of Transfer)

The SHTNAK bit specifies whether to modify PID to NAK upon the end of transfer when the selected pipe is in the receiving direction.
The SHTNAK bit is valid when the selected pipe in the receiving direction.
When the SHTNAK bit is set to 1, the USB modifies the DCPCTR.PID[1:0] bits for the DCP to NAK on determining the end of the transfer. The USB determines that the transfer has ended on the following condition.

- A short packet (including a zero-length packet) is successfully received.

## 25.2.24 DCP Maximum Packet Size Register (DCPMAXP)

Address(es): 000A 005Eh

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | DEVSEL[3:0] | | | | — | — | — | — | — | MXPS[6:0] | | | | | | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b6 to b0 | MXPS[6:0] | Maximum Packet Size*1 | These bits set the maximum amount of data (maximum packet size) in payloads for the DCP.<br>b6 b0<br>0 0 0 1 0 0 0: 8 bytes<br>0 0 1 0 0 0 0: 16 bytes<br>0 0 1 1 0 0 0: 24 bytes<br>0 1 0 0 0 0 0: 32 bytes<br>0 1 0 1 0 0 0: 40 bytes<br>0 1 1 0 0 0 0: 48 bytes<br>0 1 1 0 0 0 0: 56 bytes<br>1 0 0 0 0 0 0: 64 bytes<br>1 0 0 1 0 0 0: 72 bytes<br>1 0 1 0 0 0 0: 80 bytes<br>1 0 1 1 0 0 0: 88 bytes<br>1 1 0 0 0 0 0: 96 bytes<br>1 1 0 1 0 0 0: 104 bytes<br>1 1 1 0 0 0 0: 112 bytes<br>1 1 1 1 0 0 0: 120 bytes<br>Other than above: Setting prohibited | R/W |
| b11 to b7 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |
| b15 to b12 | DEVSEL[3:0] | Device Select*2 | b15 b12<br>0 0 0 0: Address 0000<br>0 0 0 1: Address 0001<br>0 0 1 0: Address 0010<br>0 0 1 1: Address 0011<br>0 1 0 0: Address 0100<br>0 1 0 1: Address 0101<br>Other than above: Setting prohibited | R/W |

Note 1. Modify the MXPS[6:0] bits while PID is NAK. Before modifying these bits after modifying the DCPCTR.PID[1:0] bits for the DCP from BUF to NAK, check that the DCPCTR.PBUSY bit is 0. However, if the PID[1:0] bits have been modified to NAK by the USB, checking the PBUSY bit through software is not necessary. After modifying the MXPS[6:0] bits and the DCP has been set to the CURPIPE[3:0] bits in a port select register, clear the buffer by setting the BCLR bit the port control register to 1.

Note 2. Modify the DEVSEL[3:0] bits while PID is NAK and the DCPCTR.SUREQ bit is 0. To modify these bits after modifying the DCPCTR.PID[1:0] bits for the DCP from BUF to NAK, check that the DCPCTR.PBUSY bit is 0. However, if the PID[1:0] bits have been modified to NAK by the USB, checking the PBUSY bit through software is not necessary.

### MXPS[6:0] Bits (Maximum Packet Size)

The MXPS[6:0] bits specify the maximum amount of data (maximum packet size) in payloads for the DCP. The initial value of the bits is 40h (64 bytes).

Ensure that the setting of the MXPS[6:0] bits is in compliance with USB Specifications 2.0.

Do not write to the FIFO buffer or set PID = BUF while the setting of the MXPS[6:0] bits is 0.

### DEVSEL[3:0] Bits (Device Select)

When the host controller is selected, these bits specify the address of the peripheral device which is the communication target during control transfer.

The DEVSEL[3:0] bits should be set after setting the address to the DEVADDn (n = 0 to 5) register corresponding to the value to be set in the DEVSEL[3:0] bits. For example, before setting the DEVSEL[3:0] bits to 0010b, the address should be set to DEVADD2.

When the function controller is selected, the DEVSEL[3:0] bits should be set to 0000b.

## 25.2.25 DCP Control Register (DCPCTR)

Address(es): 000A 0060h

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | BSTS | SUREQ | — | — | SUREQCLR | — | — | SQCLR | SQSET | SQMON | PBUSY | — | — | CCPL | PID[1:0] | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b1, b0 | PID[1:0] | Response PID | b1 b0<br>0 0: NAK response<br>0 1: BUF response (depending on the buffer state)<br>1 0: STALL response<br>1 1: STALL response | R/W |
| b2 | CCPL | Control Transfer End Enable | 0: Invalid<br>1: Completion of control transfer is enabled. | R/W |
| b4, b3 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |
| b5 | PBUSY | Pipe Busy | 0: DCP is not used for the transaction.<br>1: DCP is used for the transaction. | R |
| b6 | SQMON | Sequence Toggle Bit Monitor | 0: DATA0<br>1: DATA1 | R |
| b7 | SQSET | Sequence Toggle Bit Set*3 | 0: Invalid<br>1: Specifies DATA1. | R/W<br>*1 |
| b8 | SQCLR | Sequence Toggle Bit Clear*3 | 0: Invalid<br>1: Specifies DATA0. | R/W<br>*1 |
| b10, b9 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |
| b11 | SUREQCLR | SUREQ Bit Clear | 0: Invalid<br>1: Clears the SUREQ bit to 0. | R/W<br>*2 |
| b13, b12 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |
| b14 | SUREQ | Setup Token Transmission | 0: Invalid<br>1: Transmits the setup packet. | R/W<br>*2 |
| b15 | BSTS | Buffer Status | 0: Buffer access is disabled.<br>1: Buffer access is enabled. | R |

Note 1. This bit is read as 0. Only 1 can be written.
Note 2. Only 1 can be written.
Note 3. Write 1 to the SQSET and SQCLR bits while PID is NAK. Before modifying these bits after modifying the PID[1:0] bits for the DCP from BUF to NAK, check that the PBUSY bit is 0. However, if the PID[1:0] bits have been modified to NAK by the USB, checking the PBUSY bit through software is not necessary.

### PID[1:0] Bits (Response PID)

The PID[1:0] bits control the response type of the USB during control transfer.

(1) When the host controller is selected

Modify the setting of the PID[1:0] bits from NAK to BUF using the following procedure.

- When the transmitting direction is set
  Write all the transmit data to the FIFO buffer while the DVSTCTR0.UACT bit is 1 and PID is NAK, and then write 01b (BUF response). After PID has been set to BUF, the USB executes the OUT transaction.
- When the receiving direction is set
  Check that the FIFO buffer is empty (or empty the buffer) while the DVSTCTR0.UACT bit is 1 and PID is NAK, and then set PID to BUF. After PID has been set to BUF, the USB executes the IN transaction.

The USB modifies the setting of the PID[1:0] bits as follows.

- The USB sets PID to STALL (11b) on receiving the data of a size exceeding the maximum packet size when the PID[1:0] bits has been set to BUF by software.
- The USB sets PID to NAK on detecting a receive error, such as a CRC error, three consecutive times.
- The USB also sets PID to STALL (11b) on receiving the STALL handshake.

(2) When the function controller is selected

The USB modifies the setting of the PID[1:0] bits as follows.

- The USB modifies the PID[1:0] bits to NAK on receiving the setup packet. Here, the USB sets the INTSTS0.VALID bit to 1. The setting of the PID[1:0] bits cannot be modified until the VALID bit is set to 0 by software.
- The USB sets PID to STALL (11b) on receiving the data of a size exceeding the maximum packet size when the PID[1:0] bits have been set to BUF by software.
- The USB sets PID to STALL (1xb) on detecting the control transfer sequence error.
- The USB sets PID to NAK on detecting the USB bus reset.

The USB does not check to the setting of the PID[1:0] bits while the SET_ADDRESS request is processed.
The PID[1:0] bits are initialized by a USB bus reset.

#### CCPL Bit (Control Transfer End Enable)

When the function controller is selected, setting the CCPL bit to 1 enables the status stage of the control transfer to be completed.
When the CCPL bit is set to 1 by software while the corresponding PID[1:0] bits are set to BUF, the USB completes the control transfer status stage.
During control read transfer, the USB transmits the ACK handshake in response to the OUT transaction from the USB host, and transmits the zero-length packet in response to the IN transaction from the USB host during control write or no-data control transfer. However, on detecting the SET_ADDRESS request, the USB operates in auto response mode from the setup stage up to the status stage completion irrespective of the setting of the CCPL bit.
The USB modifies the CCPL bit from 1 to 0 on receiving a new setup packet.
The CCPL bit cannot be written to 1 by software while the INTSTS0.VALID bit is 1.
The CCPL bit is initialized by a USB bus reset.
When the host controller is selected, be sure to write 0 to the CCPL bit.

#### PBUSY Bit (Pipe Busy)

The PBUSY bit indicates whether DCP is used or not for the transaction when USB changes the PID[1:0] bits from BUF to NAK.
The USB modifies the PBUSY bit from 0 to 1 upon start of the USB transaction for the relevant pipe, and modifies the PBUSY bit from 1 to 0 upon completion of one transaction.
Reading the PBUSY bit after PID has been set to NAK by software allows checking whether modification of the pipe settings is possible.
For details, refer to section 25.3.4.1, Pipe Control Register Switching Procedures.

### SQMON Bit (Sequence Toggle Bit Monitor)

The SQMON bit indicates the expected value of the sequence toggle bit for the next transaction during the DCP transfer. The USB allows the SQMON bit to toggle upon normal completion of the transaction. However, the SQMON bit is not allowed to toggle when a DATA-PID mismatch occurs during the transfer in the receiving direction.
When the function controller is selected, the USB sets the SQMON bit to 1 (specifies DATA1 as the expected value) upon successful reception of the setup packet.
When the function controller is selected, the USB does not reference the SQMON bit during the IN/OUT transaction of the status stage, and does not allow the SQMON bit to toggle upon normal completion.

### SQSET Bit (Sequence Toggle Bit Set)

The SQSET bit specifies DATA1 as the expected value of the sequence toggle bit for the next transaction during the DCP transfer.
Do not set the SQCLR and SQSET bits to 1 simultaneously.

### SQCLR Bit (Sequence Toggle Bit Clear)

The SQCLR bit specifies DATA0 as the expected value of the sequence toggle bit for the next transaction during the DCP transfer. The SQCLR bit indicates 0.
Do not set the SQCLR and SQSET bits to 1 simultaneously.

### SUREQCLR Bit (SUREQ Bit Clear)

When the host controller is selected, setting the SUREQCLR bit to 1 clears the SUREQ bit to 0. The SUREQCLR bit always indicates 0.
Set the SUREQCLR bit to 1 through software when communication has stopped with the SUREQ bit being 1 during the setup transaction. However, for normal setup transactions, the USB automatically clears the SUREQ bit to 0 upon completion of the transaction; therefore, clearing the SUREQ bit through software is not necessary.
Controlling the SUREQ bit through the SUREQCLR bit must be done while the DVSTCTR0.UACT bit is 0 and thus communication is halted or while no transfer is being performed with bus disconnection detected.
When the function controller is selected, be sure to write 0 to the SUREQCLR bit.

### SUREQ Bit (Setup Token Transmission)

The USB transmits the setup packet by setting the SUREQ bit to 1 when the host controller is selected.
After completing the setup transaction process, the USB generates either the SACK or SIGN interrupt and clears the SUREQ bit to 0.
The USB also clears the SUREQ bit to 0 when software sets the SUREQCLR bit to 1.
Before setting the SUREQ bit to 1, set the DCPMAXP.DEVSEL[3:0] bits, USBREQ, USBVAL, USBINDX, and USBLENG appropriately to transmit the desired USB request in the setup transaction. Before setting this bit to 1, check that the PID[1:0] bits for the DCP are set to NAK. After setting the SUREQ bit to 1, do not modify the DCPMAXP.DEVSEL[3:0] bits, USBREQ, USBVAL, USBINDX, or USBLENG until the setup transaction is completed (the SUREQ bit = 1).
Write 1 to the SUREQ bit only when transmitting the setup token; for other purposes, write 0.
When the function controller is selected, be sure to write 0 to the SUREQ bit.

### BSTS Bit (Buffer Status)

Indicates whether DCP FIFO buffer access is enabled or disabled.
The meaning of the BSTS bit depends on the setting of ISEL bit in the port select register as shown below.

- When the ISEL bit = 0, the BSTS bit indicates whether the received data can be read from the buffer.
- When the ISEL bit = 1, the BSTS bit indicates whether the data to be transmitted can be written to the buffer.

## 25.2.26 Pipe Window Select Register (PIPESEL)

Address(es): 000A 0064h

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | PIPESEL[3:0] | | | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b3 to b0 | PIPESEL[3:0] | Pipe Window Select | b3 b0<br>0 0 0 0: No pipe selected<br>0 0 0 1: PIPE1<br>0 0 1 0: PIPE2<br>0 0 1 1: PIPE3<br>0 1 0 0: PIPE4<br>0 1 0 1: PIPE5<br>0 1 1 0: PIPE6<br>0 1 1 1: PIPE7<br>1 0 0 0: PIPE8<br>1 0 0 1: PIPE9<br>Other than above: Setting prohibited | R/W |
| b15 to b4 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |

PIPE1 to PIPE 9 should be set using PIPESEL, PIPECFG, PIPEMAXP, PIPEPERI, PIPEnCTR, PIPEnTRE, and PIPEnTRN.

After selecting the pipe using the PIPESEL register, functions of the pipe should be set using PIPECFG, PIPEMAXP, and PIPEPERI. PIPEnCTR, PIPEnTRE, and PIPEnTRN can be set regardless of the pipe selection in the PIPESEL register.

**PIPESEL[3:0] Bits (Pipe Window Select)**

The PIPESEL[3:0] bits select the pipe number corresponding to PIPECFG, PIPEMAXP, and PIPEPERI which data are written to or read from.

Selecting a pipe number through the PIPESEL[3:0] bits allows writing to and reading from PIPECFG, PIPEMAXP, and PIPEPERI which correspond to the selected pipe number.

When PIPESEL[3:0] = 0000b, 0 is read from all of the bits in PIPECFG, PIPEMAXP, and PIPEPERI. Writing to these bits is invalid.

## 25.2.27 Pipe Configuration Register (PIPECFG)

Address(es): 000A 0068h

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | TYPE[1:0] | | — | — | — | BFRE | DBLB | — | SHTNAK | — | — | DIR | EPNUM[3:0] | | | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b3 to b0 | EPNUM[3:0] | Endpoint Number*1 | These bits specify the endpoint number for the selected pipe.<br>Setting 0000b means unused pipe. | R/W |
| b4 | DIR | Transfer Direction*2,*3 | 0: Receiving direction<br>1: Transmitting direction | R/W |
| b6, b5 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |
| b7 | SHTNAK | Pipe Disabled at End of Transfer*1 | 0: Pipe assignment continued at the end of transfer<br>1: Pipe assignment disabled at the end of transfer | R/W |
| b8 | — | Reserved | This bit is read as 0. The write value should be 0. | R/W |
| b9 | DBLB | Double Buffer Mode*2,*3 | 0: Single buffer<br>1: Double buffer | R/W |
| b10 | BFRE | BRDY Interrupt Operation Specification*2,*3 | 0: BRDY interrupt upon transmitting or receiving data<br>1: BRDY interrupt upon completion of reading data | R/W |
| b13 to b11 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |
| b15, b14 | TYPE[1:0] | Transfer Type*1 | • PIPE1 and PIPE2<br>b15 b14<br>0 0: Pipe not used<br>0 1: Bulk transfer<br>1 0: Setting prohibited<br>1 1: Isochronous transfer<br><br>• PIPE3 to PIPE5<br>b15 b14<br>0 0: Pipe not used<br>0 1: Bulk transfer<br>1 0: Setting prohibited<br>1 1: Setting prohibited<br><br>• PIPE6 to PIPE9<br>b15 b14<br>0 0: Pipe not used<br>0 1: Setting prohibited<br>1 0: Interrupt transfer<br>1 1: Setting prohibited | R/W |

Note 1. Modify the TYPE[1:0], SHTNAK, and EPNUM[3:0] bits while PID is NAK. Before modifying these bits after modifying the PIPEnCTR.PID[1:0] bits for the selected pipe from BUF to NAK, check that the PIPEnCTR.PBUSY bit is 0. However, if the PID[1:0] bits have been modified to NAK by the USB, checking the PBUSY bit through software is not necessary.

Note 2. Modify the BFRE, DBLB, and DIR bits while PID is NAK and before the pipe is selected by the CURPIPE[3:0] bits in the port select register. Before modifying these bits after modifying the PIPEnCTR.PID[1:0] bits for the selected pipe from BUF to NAK, check that the PIPEnCTR.PBUSY bit is 0. However, if the PID[1:0] bits have been modified to NAK by the USB, checking the PIPEnCTR.PBUSY bit through software is not necessary.

Note 3. To modify the BFRE, DBLB, and DIR bits after completing USB communication using the selected pipe, write 1 and then 0 to the PIPEnCTR.ACLRM bit continuously through software to clear the FIFO buffer assigned to the selected pipe while the PID and CURPIPE[3:0] bits are in the state described in the above note 2.

PIPECFG specifies the transfer type, buffer memory access direction, and endpoint numbers for PIPE1 to PIPE9. It also selects single or double buffer mode, and whether to continue or disable pipe operation at the end of transfer.

#### EPNUM[3:0] Bits (Endpoint Number)

The EPNUM[3:0] bits specify the endpoint number for the selected pipe.
Setting 0000b means unused pipe.
Do not make the settings such that the combination of the settings of the DIR and EPNUM[3:0] bits should be the same for two or more pipes (bits EPNUM[3:0] = 0000b can be set for all of the pipes).

#### DIR Bit (Transfer Direction)

The DIR bit specifies the transfer direction for the selected pipe.
When the DIR bit has been set to 0 by software, the USB uses the selected pipe in the receiving direction, and when software has set the DIR bit to 1, the USB uses the selected pipe in the transmitting direction.

#### SHTNAK Bit (Pipe Disabled at End of Transfer)

The SHTNAK bit specifies whether to modify PID to NAK upon the end of transfer when the selected pipe is in the receiving direction.
The SHTNAK bit is valid when the selected pipe is PIPE1 to PIPE5 in the receiving direction.
When the SHTNAK bit has been set to 1 by software for the selected pipe in the receiving direction, the USB modifies the PIPEnCTR.PID[1:0] bits corresponding to the selected pipe to NAK on determining the end of the transfer. The USB determines that the transfer has ended on any of the following conditions.

- A short packet (including a zero-length packet) is successfully received.
- The transaction counter is used and the number of packets specified by the counter are successfully received.

#### DBLB Bit (Double Buffer Mode)

The DBLB bit selects either single or double buffer mode for the FIFO buffer used by the selected pipe.
The DBLB bit is valid when PIPE1 to PIPE5 are selected.

#### BFRE Bit (BRDY Interrupt Operation Specification)

Specifies the BRDY interrupt generation timing from the USB to the CPU with respect to the selected pipe.
When the BFRE bit has been set to 1 by software and the selected pipe is in the receiving direction, the USB detects the transfer completion and generates the BRDY interrupt on having read the relevant packet.
When the BRDY interrupt is generated with the above conditions, the BCLR bit in the port control register needs to be written to 1 by software. The FIFO buffer assigned to the selected pipe is not enabled for reception until 1 is written to the BCLR bit.
When the BFRE bit has been set to 1 by software and the selected pipe is in the transmitting direction, the USB does not generate the BRDY interrupt.
For details, refer to section 25.3.3.1, BRDY Interrupt.

#### TYPE[1:0] Bits (Transfer Type)

The TYPE[1:0] bits select the transfer type for the pipe selected by the PIPESEL.PIPESEL[3:0] bits (selected pipe).
Before setting PID to BUF for the selected pipe (before starting USB communication using the selected pipe), set the TYPE[1:0] bits to a value other than 00b.

## 25.2.28 Pipe Maximum Packet Size Register (PIPEMAXP)

Address(es): 000A 006Ch

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | DEVSEL[3:0] | | | | — | — | — | MXPS[8:0] | | | | | | | | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0/1 *1 | 0 | 0 | 0 | 0 | 0 | 0 |

Note 1. The value of these bits is 0000h when no pipe is selected with the PIPESEL.PIPESEL[3:0] bits and 0040h when a pipe is selected.

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b8 to b0 | MXPS[8:0] | Maximum Packet Size*1 | • PIPE1 and PIPE2:<br>1 byte (001h) to 256 bytes (100h)<br>• PIPE3 to PIPE5:<br>8 bytes (008h), 16 bytes (010h),<br>32 bytes (020h), 64 bytes (040h)<br>(Bits [8:7] and [2:0] are not provided.)<br>• PIPE6 to PIPE9:<br>1 byte (001h) to 64 bytes (040h)<br>(Bits [8:7] are not provided.) | R/W |
| b11 to b9 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |
| b15 to b12 | DEVSEL[3:0] | Device Select*2 | b3 b0<br>0 0 0 0: Address 0000<br>0 0 0 1: Address 0001<br>0 0 1 0: Address 0010<br>0 0 1 1: Address 0011<br>0 1 0 0: Address 0100<br>0 1 0 1: Address 0101<br>Other than above: Setting prohibited | R/W |

Note 1. Modify the MXPS[8:0] bits while PID is NAK and before the pipe is selected by the CURPIPE[3:0] bits in the port select register. Before modifying these bits after modifying the PIPEnCTR.PID[1:0] bits for the selected pipe from BUF to NAK, check that the PIPEnCTR.PBUSY bit is 0. However, if the PID[1:0] bits have been modified to NAK by the USB, checking the PBUSY bit through software is not necessary.

Note 2. Modify the DEVSEL[3:0] bits while PID is NAK. To modify these bits after modifying the PIPEnCTR.PID[1:0] bits for the selected pipe from BUF to NAK, check that the PIPEnCTR.PBUSY bit is 0. However, if the PID[1:0] bits have been modified to NAK by the USB, checking the PBUSY bit through software is not necessary.

PIPEMAXP specifies the maximum packet size for PIPE1 to PIPE9.

### MXPS[8:0] Bits (Maximum Packet Size)

Specifies the maximum data payload (maximum packet size) for the selected pipe.

These bits should be set to the appropriate value for each transfer type based on USB Specifications 2.0. While MXPS[8:0] = 0, do not write to the FIFO buffer or set PID to BUF.

### DEVSEL[3:0] Bits (Device Select)

When the host controller is selected, these bits specify the USB device address of the peripheral device which is the communication target.

The DEVSEL[3:0] bits should be set after setting the address to the DEVADDn (n = 0 to 5) register corresponding to the value to be set in the DEVSEL[3:0] bits. For example, before setting the DEVSEL[3:0] bits to 0010b, the address should be set to DEVADD2.

When the function controller is selected, the DEVSEL[3:0] bits should be set to 0000b.

## 25.2.29 Pipe Cycle Control Register (PIPEPERI)

Address(es): 000A 006Eh

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | IFIS | — | — | — | — | — | — | — | — | — | IITV[2:0] | | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b2 to b0 | IITV[2:0] *1 | Interval Error Detection Interval | Specifies the interval error detection timing for the selected pipe in terms of frames, which is expressed as nth power of 2. | R/W |
| b11 to b3 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |
| b12 | IFIS | Isochronous IN Buffer Flush | 0: The buffer is not flushed.<br>1: The buffer is flushed. | R/W |
| b15 to b13 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |

Note 1. Modify the IITV[2:0] bits while PID is NAK. To modify these bits after modifying the PIPEnCTR.PID[1:0] bits for the selected pipe from BUF to NAK, check that the PIPEnCTR.PBUSY bit is 0. However, if the PID[1:0] bits have been modified to NAK by the USB, checking the PBUSY bit through software is not necessary.

PIPEPERI selects whether the buffer is flushed or not when an interval error occurred during isochronous IN transfer, and sets the interval error detection interval for PIPE1 to PIPE9.

### IITV[2:0] Bits (Interval Error Detection Interval)

Before modifying the IITV[2:0] bits after USB communication has been completed with the IITV[2:0] bits set to a certain value, set PID to NAK and then set the PIPEnCTR.ACLRM bit to 1 to initialize the interval timer.
The IITV[2:0] bits are invalid for PIPE3 to PIPE5; set the IITV[2:0] bits to 000b for PIPE3 to PIPE5.

### IFIS Bit (Isochronous IN Buffer Flush)

Specifies whether to flush the buffer when the pipe selected by the PIPESEL.PIPESEL[3:0] bits (selected pipe) is used for isochronous IN transfers.
When the function controller is selected and the selected pipe is for isochronous IN transfers, the USB automatically clears the FIFO buffer when the USB fails to receive the IN token from the USB host within the interval set by the IITV[2:0] bits in terms of frames.
In double buffer mode (the PIPECFG.DBLB bit = 1), the USB only clears the data in the plane used earlier.
The USB clears the FIFO buffer on receiving the SOF packet immediately after the frame in which the USB has expected to receive the IN token. Even if the SOF packet is damaged, the USB also clears the FIFO buffer at the right timing to receive the SOF packet by using the internal interpolation function.
When the host controller is selected, set the IITV[2:0] bits to 000b.
When the selected pipe is not for isochronous transfer, set the IITV[2:0] bits to 000b.

## 25.2.30 PIPEn Control Registers (PIPEnCTR) (n = 1 to 9)

- PIPEnCTR (n = 1 to 5)

Address(es): PIPE1CTR: 000A 0070h, PIPE2CTR: 000A 0072h, PIPE3CTR: 000A 0074h, PIPE4CTR: 000A 0076h, PIPE5CTR: 000A 0078h

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | BSTS | INBUFM | — | — | — | ATREPM | ACLRM | SQCLR | SQSET | SQMON | PBUSY | — | — | — | PID[1:0] | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b1, b0 | PID[1:0] | Response PID | b1 b0<br>0 0: NAK response<br>0 1: BUF response (depending on the buffer state)<br>1 0: STALL response<br>1 1: STALL response | R/W |
| b4 to b2 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |
| b5 | PBUSY | Pipe Busy | 0: The relevant pipe is not used for the transaction.<br>1: The relevant pipe is used for the transaction. | R |
| b6 | SQMON | Sequence Toggle Bit Confirmation | 0: DATA0<br>1: DATA1 | R |
| b7 | SQSET | Sequence Toggle Bit Set*2 | 0: Invalid<br>1: Specifies DATA1. | R/W *1 |
| b8 | SQCLR | Sequence Toggle Bit Clear*2 | 0: Invalid<br>1: Specifies DATA0. | R/W *1 |
| b9 | ACLRM | Auto Buffer Clear Mode*3 | 0: Disabled<br>1: Enabled (all buffers are initialized) | R/W |
| b10 | ATREPM | Auto Response Mode*2 | 0: Auto response is disabled.<br>1: Auto response is enabled. | R/W |
| b13 to b11 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |
| b14 | INBUFM | Transmit Buffer Monitor | 0: There is no data to be transmitted in the buffer memory.<br>1: There is data to be transmitted in the buffer memory. | R |
| b15 | BSTS | Buffer Status | 0: Buffer access by the CPU is disabled.<br>1: Buffer access by the CPU is enabled. | R |

Note 1. Only 0 can be read. Only 1 can be written.
Note 2. Modify the ATREPM bit or write 1 to the SQCLR or SQSET bit while PID is NAK. Before modifying these bits after modifying the PID[1:0] bits for the selected pipe from BUF to NAK, check that the PBUSY bit is 0. However, if the PID[1:0] bits have been modified to NAK by the USB, checking the PBUSY bit through software is not necessary.
Note 3. Modify the ACLRM bit while PID[1:0] is NAK and before the pipe is selected by the CURPIPE[3:0] bits in the port select register. Before modifying this bit after modifying the PID[1:0] bits for the selected pipe from BUF to NAK, check that the PBUSY bit is 0. However, if the PID[1:0] bits have been modified to NAK by the USB, checking the PBUSY bit through software is not necessary.

PIPEnCTR can be set regardless of the pipe selection in the PIPESEL register.

### PID[1:0] Bits (Response PID)

The PID[1:0] bits specify the response type for the next transaction of the relevant pipe.
The default setting of the PID[1:0] bits is NAK. Modify the setting of the PID[1:0] bits to BUF to use the relevant pipe for USB transfer. Table 25.7 and Table 25.8 show the basic operation (operation when there are no errors in the transmitted and received packets) of the USB depending on the PID[1:0] bit setting.
After modifying the setting of the PID[1:0] bits through software from BUF to NAK during USB communication using the relevant pipe, check that the PBUSY bit is 1 to see if USB transfer using the relevant pipe has actually entered the NAK state. However, if the PID[1:0] bits have been modified to NAK by the USB, checking the PBUSY bit through software is not necessary.
The USB modifies the setting of the PID[1:0] bits in the following cases.

- The USB sets PID to NAK on recognizing the completion of the transfer when the relevant pipe is in the receiving direction and the PIPECFG.SHTNAK bit for the selected pipe has been set to 1 by software.
- The USB sets PID to STALL (11b) on receiving a data packet with a payload exceeding the maximum packet size of the relevant pipe.
- The USB sets PID to NAK on detecting a USB bus reset when the function controller is selected.
- The USB sets PID to NAK on detecting a receive error, such as a CRC error, three consecutive times when the host controller is selected.
- The USB sets PID to STALL (11b) on receiving the STALL handshake when the host controller is selected.

To specify each response type, set the PID[1:0] bits as follows.

- To make a transition from NAK (00b) to STALL, set 10b.
- To make a transition from BUF (01b) to STALL, set 11b.
- To make a transition from STALL (11b) to NAK, set 10b and then 00b.
- To make a transition from STALL to BUF, set 00b (NAK) and then 01b (BUF).

### PBUSY Bit (Pipe Busy)

The PBUSY bit indicates whether the relevant pipe is being currently used or not for the transaction.
The USB modifies the PBUSY bit from 0 to 1 upon start of the USB transaction for the relevant pipe, and modifies the PBUSY bit from 1 to 0 upon completion of one transaction.
Reading the PBUSY bit after PID to NAK has been set to NAK by software allows checking whether modification of the pipe settings is possible.
For details, refer to section 25.3.4.1, Pipe Control Register Switching Procedures.

### SQMON Bit (Sequence Toggle Bit Confirmation)

The SQMON bit indicates the expected value of the sequence toggle bit for the next transaction of the relevant pipe.
When the relevant pipe is not for the isochronous transfer, the USB allows the SQMON bit to toggle upon normal completion of the transaction. However, the SQMON bit is not allowed to toggle when a DATA-PID mismatch occurs during the transfer in the receiving direction.

### SQSET Bit (Sequence Toggle Bit Set)

The SQSET bit should be set to 1 to set DATA1 as the expected value of the sequence toggle bit for the next transaction of the relevant pipe.
Setting the SQSET bit to 1 through software allows the USB to set DATA1 as the expected value of the sequence toggle bit of the relevant pipe. The USB sets the SQSET bit to 0.

### SQCLR Bit (Sequence Toggle Bit Clear)

The SQCLR bit should be set to 1 to clear the expected value (to set DATA0 as the expected value) of the sequence toggle bit for the next transaction of the relevant pipe.
Setting the SQCLR bit to 1 through software allows the USB to set DATA0 as the expected value of the sequence toggle bit of the relevant pipe. The USB sets the SQCLR bit to 0.

### ACLRM Bit (Auto Buffer Clear Mode)

Enables or disables auto buffer clear mode for the relevant pipe.
To delete the information in the FIFO buffer assigned to the relevant pipe completely, write 1 and then 0 to the ACLRM bit continuously.
Table 25.9 shows the information cleared by writing 1 and 0 to the ACLRM bit continuously and the cases in which clearing the information is necessary.

### ATREPM Bit (Auto Response Mode)

Enables or disables auto response mode for the relevant pipe.
When the function controller is selected and the relevant pipe is for bulk transfer, the ATREPM bit can be set to 1.
When the ATREPM bit is set to 1, the USB responds to the token from the USB host as described below.
(1) When the relevant pipe is for bulk IN transfer (the PIPECFG.TYPE[1:0] bits = 01b and the PIPECFG.DIR bit = 1)
When the ATREPM bit = 1 and PID = BUF, the USB transmits a zero-length packet in response to the IN token.
The USB updates (allows toggling of) the sequence toggle bit (DATA-PID) each time the USB receives ACK from the USB host (in a single transaction, IN token is received, zero-length packet is transmitted, and then ACK is received.).
In this case, the USB does not generate the BRDY or BEMP interrupt.
(2) When the relevant pipe is for bulk OUT transfer (the PIPECFG.TYPE[1:0] bits = 01b and the PIPECFG.DIR bit = 0)
When the ATREPM bit = 1 and PID = BUF, the USB returns NAK in response to the OUT token and generates the NRDY interrupt.
For USB communication in auto response mode, set the ATREPM bit to 1 while the FIFO buffer is empty. Do not write to the FIFO buffer during USB communication in auto response mode.
When the relevant pipe is for isochronous transfer, be sure to set the ATREPM bit to 0.
When the host controller is selected, be sure to set the ATREPM bit to 0.

### INBUFM Bit (Transmit Buffer Monitor)

Indicates the relevant FIFO buffer status when the relevant pipe is in the transmitting direction.
When the relevant pipe is in the transmitting direction (the PIPECFG.DIR bit = 1), the USB sets the INBUFM bit to 1 when the CPU or DTC completes writing data to at least one FIFO buffer plane.
The USB sets the INBUFM bit to 0 when the USB completes transmitting the data from the FIFO buffer plane to which all the data has been written. In double buffer mode (the PIPECFG.DBLB bit = 1), the USB sets the INBUFM bit to 0 when the USB completes transmitting the data from the two FIFO buffer planes before the CPU or DTC completes writing data to one FIFO buffer plane.
The INBUFM bit indicates the same value as the BSTS bit when the relevant pipe is in the receiving direction (the PIPECFG.DIR bit = 0).

### BSTS Bit (Buffer Status)

Indicates the FIFO buffer status for the relevant pipe.

The meaning of the BSTS bit depends on the settings of the PIPECFG.DIR bit, PIPECFG.BFRE bit, and DnFIFOSEL.DCLRM bits as shown in Table 25.10.

**Table 25.7 Operation of USB depending on PID[1:0] Bits Setting (When Host Controller is Selected)**

| Bits PID[1:0] | Transfer Type | Transfer Direction (DIR Bit) | Operation of USB |
|---|---|---|---|
| 00b (NAK) | Operation does not depend on the setting. | Operation does not depend on the setting. | Does not issue tokens. |
| 01b (BUF) | Bulk or interrupt | Operation does not depend on the setting. | Issues tokens while the DVSTCTR0.UACT bit is 1 and the FIFO buffer corresponding to the relevant pipe is ready for transmission and reception.<br>Does not issue tokens while the DVSTCTR0.UACT bit is 0 or the FIFO buffer corresponding to the relevant pipe is not ready for transmission or reception. |
| | Isochronous | Operation does not depend on the setting. | Issues tokens irrespective of the status of the FIFO buffer corresponding to the relevant pipe. |
| 10b (STALL) or 11b (STALL) | Operation does not depend on the setting. | Operation does not depend on the setting. | Does not issue tokens. |

**Table 25.8 Operation of USB depending on PID[1:0] Bits Setting (When Function Controller is Selected)**

| Bits PID[1:0] | Transfer Type | Transfer Direction (DIR Bit) | Operation of USB |
|---|---|---|---|
| 00b (NAK) | Bulk or interrupt | Operation does not depend on the setting. | Returns NAK in response to the token from the USB host. |
| | Isochronous | Operation does not depend on the setting. | Returns nothing in response to the token from the USB host. |
| 01b (BUF) | Bulk | Receiving direction (DIR bit = 0) | Receives data and returns ACK in response to the OUT token from the USB host if the FIFO buffer corresponding to the relevant pipe is ready for reception. |
| | Interrupt | Receiving direction (DIR bit = 0) | Receives data and returns ACK in response to the OUT token from the USB host if the FIFO buffer corresponding to the relevant pipe is ready for reception. |
| | Bulk or interrupt | Transmitting direction (DIR bit = 1) | Transmits data in response to the token from the USB host if the corresponding FIFO buffer is ready for transmission. Returns NAK if not ready. |
| | Isochronous | Receiving direction (DIR bit = 0) | Receives data in response to the OUT token from the USB host if the FIFO buffer corresponding to the relevant pipe is ready for reception. Discards data if not ready. |
| | Isochronous | Transmitting direction (DIR bit = 1) | Transmits data in response to the token from the USB host if the corresponding FIFO buffer is ready for transmission. Transmits the zero-length packet if not ready. |
| 10b (STALL) or 11b (STALL) | Bulk or interrupt | Operation does not depend on the setting. | Returns STALL in response to the token from the USB host. |
| | Isochronous | Operation does not depend on the setting. | Returns nothing in response to the token from the USB host. |

**Table 25.9 Information Cleared by USB by Setting ACLRM = 1**

| No. | Information Cleared by ACLRM Bit Manipulation | Cases in which Clearing Information is Necessary |
|---|---|---|
| 1 | All the information in the FIFO buffer assigned to the relevant pipe (both FIFO buffer planes are cleared when double buffer mode is selected) | When the pipe is to be initialized |
| 2 | The interval count value when the relevant pipe is for isochronous transfer | When the interval count value is to be reset |
| 3 | Internal flags concerning the PIPECFG.BFRE bit | When the PIPECFG.BFRE setting is modified |
| 4 | FIFO buffer toggle control | When the PIPECFG.DBLB setting is modified |
| 5 | Internal flags concerning the transaction count | When the transaction count function is forcibly terminated |

**Table 25.10 Operation of BSTS Bit**

| DIR Bit | BFRE Bit | DCLRM Bit | BSTS Bit Function |
|---|---|---|---|
| 0 | 0 | 0 | The received data can be read from the FIFO buffer.<br>The received data has been completely read from the FIFO buffer. |
| | | 1 | Setting prohibited |
| | 1 | 0 | The received data can be read from the FIFO buffer.<br>The BCLR bit in the port control register has been set to 1 by software after the received data has been completely read from the FIFO buffer. |
| | | 1 | The received data can be read from the FIFO buffer.<br>The received data has been completely read from the FIFO buffer. |
| 1 | 0 | 0 | The transmit data can be written to the FIFO buffer.<br>The transmit data has been completely written to the FIFO buffer. |
| | | 1 | Setting prohibited |
| | 1 | 0 | Setting prohibited |
| | | 1 | Setting prohibited |

- PIPEnCTR (n = 6 to 9)

Address(es): PIPE6CTR: 000A 007Ah, PIPE7CTR: 000A 007Ch, PIPE8CTR: 000A 007Eh, PIPE9CTR: 000A 0080h

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | BSTS | — | — | — | — | — | ACLRM | SQCLR | SQSET | SQMON | PBUSY | — | — | — | PID[1:0] | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b1, b0 | PID[1:0] | Response PID | b1 b0<br>0 0: NAK response<br>0 1: BUF response (depending on the buffer state)<br>1 0: STALL response<br>1 1: STALL response | R/W |
| b4 to b2 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |
| b5 | PBUSY | Pipe Busy | 0: The relevant pipe is not used at the USB bus.<br>1: The relevant pipe is used at the USB bus. | R |
| b6 | SQMON | Sequence Toggle Bit Confirmation | 0: DATA0<br>1: DATA1 | R |
| b7 | SQSET | Sequence Toggle Bit Set*2 | 0: Invalid<br>1: Specifies DATA1. | R/W *1 |
| b8 | SQCLR | Sequence Toggle Bit Clear*2 | 0: Invalid<br>1: Specifies DATA0. | R/W *1 |
| b9 | ACLRM | Auto Buffer Clear Mode*2,*3 | 0: Auto buffer clear mode is disabled.<br>1: Auto buffer clear mode is enabled (all buffers are initialized) | R/W |
| b14 to b10 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |
| b15 | BSTS | Buffer Status | 0: Buffer access is disabled.<br>1: Buffer access is enabled. | R |

Note 1. Only 0 can be read. Only 1 can be written.
Note 2. Write 1 to the SQCLR or SQSET bit while PID is NAK. Before modifying these bits after modifying the PID[1:0] bits for the selected pipe from BUF to NAK, check that the PBUSY bit is 0. However, if the PID[1:0] bits have been modified to NAK by the USB, checking the PBUSY bit through software is not necessary.
Note 3. Modify the ACLRM bit while PID is NAK and before the pipe is selected by the CURPIPE[3:0] bits in the port select register. Before modifying this bit after modifying the PID[1:0] bits for the selected pipe from BUF to NAK, check that the PBUSY bit is 0. However, if the PID[1:0] bits have been modified to NAK by the USB, checking the PBUSY bit through software is not necessary.

#### PID[1:0] Bits (Response PID)

The PID[1:0] bits specify the response type for the next transaction of the relevant pipe.
The default setting of the PID[1:0] bits is NAK. Modify the setting of the PID[1:0] bits to BUF to use the relevant pipe for USB transfer. Table 25.7 and Table 25.8 show the basic operation (operation when there are no errors in the transmitted and received packets) of the USB depending on the setting of the PID[1:0] bits.
After modifying the setting of the PID[1:0] bits through software from BUF to NAK during USB communication using the relevant pipe, check that the PBUSY bit is 1 to see if USB transfer using the relevant pipe has actually entered the NAK state. However, if the PID bits have been modified to NAK by the USB, checking the PBUSY bit through software is not necessary.
The USB modifies the setting of the PID[1:0] bits in the following cases.

- The USB sets PID to STALL (11b) on receiving a data packet with a payload exceeding the maximum packet size of the relevant pipe.
- The USB sets PID to NAK on detecting a USB bus reset when the function controller is selected.
- The USB sets PID to NAK on detecting a receive error, such as a CRC error, three consecutive times when the host controller is selected.
- The USB sets PID to STALL (11b) on receiving the STALL handshake when the host controller is selected.

To specify each response type, set the PID[1:0] bits as follows.

- To make a transition from NAK (00b) to STALL, set 10b.
- To make a transition from BUF (01b) to STALL, set 11b.
- To make a transition from STALL (11b) to NAK, set 10b and then 00b.
- To make a transition from STALL to BUF, set 00b (NAK) and then 01b (BUF).

#### PBUSY Bit (Pipe Busy)

The PBUSY bit indicates whether the relevant pipe is being currently used or not for the transaction.
The USB modifies the PBUSY bit from 0 to 1 upon start of the USB transaction for the relevant pipe, and modifies the PBUSY bit from 1 to 0 upon completion of one transaction.
Reading the PBUSY bit after PID has been set to NAK by software allows checking whether modification of the pipe settings is possible.

#### SQMON Bit (Sequence Toggle Bit Confirmation)

The SQMON bit indicates the expected value of the sequence toggle bit for the next transaction of the relevant pipe.
The USB allows the SQMON bit to toggle upon normal completion of the transaction. However, the SQMON bit is not allowed to toggle when a DATA-PID mismatch occurs during the transfer in the receiving direction.

#### SQSET Bit (Sequence Toggle Bit Set)

The SQSET bit should be set to 1 to set DATA1 as the expected value of the sequence toggle bit for the next transaction of the relevant pipe.
Setting the SQSET bit through software allows the USB to set DATA1 as the expected value of the sequence toggle bit of the relevant pipe. The USB sets the SQSET bit to 0.

#### SQCLR Bit (Sequence Toggle Bit Clear)

The SQCLR bit should be set to 1 to clear the expected value (to set DATA0 as the expected value) of the sequence toggle bit for the next transaction of the relevant pipe.
Setting the SQCLR bit to 1 through software allows the USB to set DATA0 as the expected value of the sequence toggle bit of the relevant pipe. The USB sets the SQCLR bit to 0.

### ACLRM Bit (Auto Buffer Clear Mode)

Enables or disables auto buffer clear mode for the relevant pipe.

To delete the information in the FIFO buffer assigned to the relevant pipe completely, write 1 and then 0 to the ACLRM bit continuously.

Table 25.11 shows the information cleared by writing 1 and 0 to the ACLRM bit continuously and the cases in which clearing the information is necessary.

### BSTS Bit (Buffer Status)

Indicates the FIFO buffer status for the relevant pipe.

The meaning of the BSTS bit depends on the settings of the PIPECFG.DIR bit, PIPECFG.BFRE bit, and DnFIFOSEL.DCLRM bits as shown in Table 25.10.

**Table 25.11 Information Cleared by USB by Setting the ACLRM Bit = 1**

| No. | Information Cleared by ACLRM Bit Manipulation | Cases in which Clearing Information is Necessary |
|---|---|---|
| 1 | All the information in the FIFO buffer assigned to the selected pipe | When the pipe is to be initialized |
| 2 | The interval count value when the selected pipe is for interrupt transfer and the host controller is selected | When the interval count value is to be reset |
| 3 | Internal flags concerning the PIPECFG.BFRE bit | When the PIPECFG.BFRE setting is modified |
| 4 | Internal flags concerning the transaction count | When the transaction count function is forcibly terminated |

## 25.2.31 PIPEn Transaction Counter Enable Register (PIPEnTRE) (n = 1 to 5)

Address(es): PIPE1TRE: 000A 0090h, PIPE2TRE: 000A 0094h, PIPE3TRE: 000A 0098h, PIPE4TRE: 000A 009Ch, PIPE5TRE: 000A 00A0h

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | TRENB | TRCLR | — | — | — | — | — | — | — | — |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b7 to b0 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |
| b8 | TRCLR | Transaction Counter Clear | 0: Invalid<br>1: The current counter value is cleared. | R/W |
| b9 | TRENB | Transaction Counter Enable | 0: Transaction counter is disabled.<br>1: Transaction counter is enabled. | R/W |
| b15 to b10 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |

Note: • Modify each bit in PIPEnTRE while PID is NAK. Before modifying these bits after modifying the PIPEnCTR.PID[1:0] bits for the selected pipe from BUF to NAK, check that the PIPEnCTR.PBUSY bit is 0. However, if the PID[1:0] bits have been modified to NAK by the USB, checking the PBUSY bit through software is not necessary.

### TRCLR Bit (Transaction Counter Clear)

Clears the current value of the transaction counter corresponding to the relevant pipe and then sets the TRCLR bit to 0.

### TRENB Bit (Transaction Counter Enable)

Enables or disables the transaction counter.

For the pipe in the receiving direction, setting the TRENB bit to 1 after setting the total number of the packets to be received in the PIPEnTRN.TRNCNT[15:0] bits through software allows the USB to control hardware as described below on having received the number of packets equal to the setting of the TRNCNT[15:0] bits.

- While the PIPECFG.SHTNAK bit is 1, the USB modifies the PID bits to NAK for the corresponding pipe on having received the number of packets equal to the setting of the TRNCNT[15:0] bits.
- While the PIPECFG.BFRE bit is 1, the USB asserts the BRDY interrupt on having received the number of packets equal to the setting of the TRNCNT[15:0] bits and then reading the last received data.

For the pipe in the transmitting direction, set the TRENB bit to 0.

When the transaction counter is not used, set the TRENB bit to 0.

When the transaction counter is used, set the TRNCNT[15:0] bits before setting the TRENB bit to 1. Set the TRENB bit to 1 before receiving the first packet to be counted by the transaction counter.

## 25.2.32 PIPEn Transaction Counter Register (PIPEnTRN) (n = 1 to 5)

Address(es): PIPE1TRN: 000A 0092h, PIPE2TRN: 000A 0096h, PIPE3TRN: 000A 009Ah, PIPE4TRN: 000A 009Eh, PIPE5TRN: 000A 00A2h

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | TRNCNT[15:0] | | | | | | | | | | | | | | | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b15 to b0 | TRNCNT[15:0] | Transaction Counter | • When written to:<br>Specifies the total of packets (number of transactions) to be received in corresponding PIPE.<br>• When read from:<br>Indicates the specified number of transactions if the PIPEnTRE.TRENB bit is 0.<br>Indicates the number of currently counted transactions if the PIPEnTRE.TRENB bit is 1. | R/W |

PIPEnTRN retains the setting by a USB bus reset.

### TRNCNT[15:0] Bits (Transaction Counter)

The USB increments the value of the TRNCNT[15:0] bits by one when all of the following conditions are satisfied on receiving the packet.

- The PIPEnTRE.TRENB bit = 1
- (TRNCNT[15:0] set value ≠ current counter value + 1) on receiving the packet.
- The payload of the received packet agrees with the setting of the PIPEMAXP.MXPS[8:0] bits.

The USB clears the value of the TRNCNT[15:0] bits to 0 when any of the following conditions are satisfied.

(1) All of the following conditions are satisfied.

- The PIPEnTRE.TRENB bit = 1
- (TRNCNT[15:0] set value = current counter value + 1) on receiving the packet.
- The payload of the received packet agrees with the setting of the PIPEMAXP.MXPS[8:0] bits.

(2) All of the following conditions are satisfied.

- The PIPEnTRE.TRENB bit = 1
- The USB has received a short packet.

(3) All of the following conditions are satisfied.

- The PIPEnTRE.TRENB bit = 1
- The PIPEnTRE.TRCLR bit has been set to 1 by software.

For the pipe in the transmitting direction, set the TRNCNT[15:0] bits to 0.
When the transaction counter is not used, set the TRNCNT[15:0] bits to 0.
Setting the number of transactions to be transferred to the TRNCNT[15:0] bits is only enabled when the PIPEnTRE.TRENB bit is 0. To modify the number of transactions to be transferred, set the TRCLR bit to 1 (to clear the current counter value) before setting the PIPEnTRE.TRENB bit to 1.

## 25.2.33 Device Address n Configuration Register (DEVADDn) (n = 0 to 5)

Address(es): DEVADD0: 000A 00D0h, DEVADD1: 000A 00D2h, DEVADD2: 000A 00D4h, DEVADD3: 000A 00D6h, DEVADD4: 000A 00D8h, DEVADD5: 000A 00DAh

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | USBSPD[1:0] | | — | — | — | — | — | — |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b5 to b0 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |
| b7, b6 | USBSPD[1:0] | Transfer Speed of Communication Target Device | b7 b6<br>0 0: DEVADDn is not used<br>0 1: Low speed<br>1 0: Full speed<br>1 1: Setting prohibited | R/W |
| b15 to b8 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |

DEVADDn specifies the transfer speed of the peripheral device which is the communication target for PIPE0 to PIPE9. When the host controller is selected, the bits in DEVADDn should be set before starting communication using each pipe. The bits in DEVADDn should be modified while no valid pipes are using the settings of the bits. Valid pipes refer to the pipes satisfying both of the following conditions:

- DEVADDn is selected by the DEVSEL[3:0] bits.
- The PID[1:0] bits are set to BUF for the selected pipe or the selected pipe is the DCP with the DCPCTR.SUREQ bit set to 1.

**USBSPD[1:0] Bits (Transfer Speed of Communication Target Device)**

Specifies the USB transfer speed of the communication target peripheral device.
Set these bits to 01b when a low-speed device is connected via the HUB, whereas set them to 10b when a full-speed device is connected.
When the host controller is selected, the USB refers to the setting of the USBSPD[1:0] bits to generate packets.
When the function controller is selected, set these bits to 00b.

## 25.2.34 USB Module Control Register (USBMC)

Address(es): 000A 00CCh

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | VDDUSBE |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b0 | VDDUSBE | USB Power Supply Circuit On/Off Control | 0: USB power supply circuit off<br>1: USB power supply circuit on | R/W |
| b1 | — | Reserved | This bit is read as 1. The write value should be 1. | R/W |
| b15 to b2 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |

**VDDUSBE Bit (USB Power Supply Circuit On/Off Control)**

The USB power supply circuit generates the reference voltage for battery charging. Set this bit to 1 before using the battery charging function.

## 25.2.35 BC Control Register 0 (USBBCCTRL0)

Address(es): 000A 00B0h

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | PDDETSTS0 | CHGDETSTS0 | BATCHGE0 | — | VDMSRCE0 | IDPSINKE0 | VDPSRCE0 | IDMSINKE0 | IDPSRCE0 | RPDME0 |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b0 | RPDME0 | D– Pin Pull-Down Control | 0: Pull-down off<br>1: Pull-down on | R/W |
| b1 | IDPSRCE0 | D+ Pin IDPSRC Output Control | 0: Stop<br>1: 10 µA output | R/W |
| b2 | IDMSINKE0 | D– Pin 0.6 V Input Detection (Comparator and Sink) Control | 0: Detection off<br>1: Detection on (comparator and sink current on) | R/W |
| b3 | VDPSRCE0 | D+ Pin VDPSRC (0.6 V) Output Control | 0: Stop<br>1: 0.6 V output | R/W |
| b4 | IDPSINKE0 | D+ Pin 0.6 V Input Detection (Comparator and Sink) Control | 0: Detection off<br>1: Detection on (comparator and sink current on) | R/W |
| b5 | VDMSRCE0 | D– Pin VDMSRC (0.6 V) Output Control | 0: Stop<br>1: 0.6 V output | R/W |
| b6 | — | Reserved | This bit is read as 0. The write value should be 0. | R/W |
| b7 | BATCHGE0 | BC (Battery Charger) Function Ch0 General Enable Control | 0: Disabled<br>1: Enabled | R/W |
| b8 | CHGDETSTS0 | D– Pin 0.6 V Input Detection Status*1 | 0: Not detected<br>1: Detected | R |
| b9 | PDDETSTS0 | D+ Pin 0.6 V Input Detection Status*2 | 0: Not detected<br>1: Detected | R |
| b15 to b10 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |

Note 1. Valid when IDMSINKE0 = 1.
Note 2. Valid when IDPSINKE0 = 1.

### RPDME0 Bit (D– Pin Pull-Down Control)

When using the battery charging function, set this bit to 1 to control the pull-down resistor of the D– pin.

### IDPSRCE0 Bit (D+ Pin IDPSRC Output Control)

With this bit set to 1, when the function controller is selected, current output is enabled upon detection of the connection of the data pin and the D+ pin is pulled up.

### IDMSINKE0 Bit (D– Pin 0.6 V Input Detection (Comparator and Sink) Control)

With this bit set to 1, when the function controller is selected, the USB module detects whether VDMSRC (0.6 V) that is output from the host to D– upon primary detection is connected, or whether VDPSRC (0.6 V) that is output from the function to D+ is connected to the function's D– via the host.

### VDPSRCE0 Bit (D+ Pin VDPSRC (0.6 V) Output Control)

With this bit set to 1, when the function controller is selected, output is enabled upon primary detection and VDPSRC (0.6 V) is applied to D+.

**IDPSINKE0 Bit (D+ Pin 0.6 V Input Detection (Comparator and Sink) Control)**

With this bit set to 1, when the function controller selected, the USB module detects whether VDMSRC (0.6 V) that is output from the function to D– is connected to the function's D+ (DCP) via the host. When the host controller is selected, the USB module detects whether VDPSRC (0.6 V) that is output from the function to D+ upon primary detection is connected.

**VDMSRCE0 Bit (D– Pin VDMSRC (0.6 V) Output Control)**

With this bit set to 1, when the function controller selected, output is enabled upon secondary detection and VDMSRC (0.6 V) is applied to D–. When the host controller is selected, output is enabled upon primary detection and VDMSRC (0.6 V) is applied to D–.

**CHGDETSTS0 Bit (D– Pin 0.6 V Input Detection Status)**

When the host controller is selected, this bit is set to 1 if the USB module detects whether VDMSRC (0.6 V) that is output from the host to D– during primary detection is connected, or whether VDPSRC (0.6 V) that is output from the function to D+ is connected to the function's D– via the host.

**PDDETSTS0 Bit (D+ Pin 0.6 V Input Detection Status)**

When the function controller is selected, this bit is set to 1 if the USB module detects whether VDMSRC (0.6 V) that is output from the function to D– during secondary detection is connected to the function's D+ (DCP) via the host.
When the host controller is selected, this bit is set to 1 if the USB module detects whether VDPSRC (0.6 V) that is output from the function to D+ during primary detection is connected.

## 25.3 Operation

### 25.3.1 System Control

This section describes the register settings that are necessary for initialization of this module and power consumption control.

#### 25.3.1.1 Setting Data to the USB Related Register

Setting the SYSCFG.USBE bit to 1 after starting the clock supply to the USB (SYSCFG.SCKE bit = 1) enables and starts USB operation.

#### 25.3.1.2 Controller Function Selection

For the USB, the host or function controller can be selected using the DCFM bit in SYSCFG. The DCFM bit should be modified in the initial settings immediately after a power-on reset or in the D+ pull-up-disabled (SYSCFG.DPRPU bit = 0) and D+ and D– pull-down-disabled (SYSCFG.DRPD bit = 0) state.

#### 25.3.1.3 Controlling USB Data Bus Resistors

The USB has pull-up and pull-down resistors for the D+ and D– lines. Pull up or pull down these lines by setting the SYSCFG.DPRPU and DRPD bits.
When the function controller is selected, confirm that connection to the USB host is made, then set the SYSCFG.DPRPU bit to 1 and pull up the D+ line (during full speed) and D– line (during low speed).
When the SYSCFG.DPRPU bit is set to 1 during communication with the PC, the USB module disables the pull-up resistor of the USB data line, thus notifying the USB host of disconnection.
When the host controller is selected, set the SYSCFG.DRPD bit and pull down the D+ and D– lines.

**Table 25.12 Controlling USB Data Bus Resistors**

| Settings | | | Controlling USB Data Bus Resistors | | |
|---|---|---|---|---|---|
| DRPD | DPRPU | DMRPU | D– | D+ | Remarks |
| 0 | 0 | 0 | Open | Open | When not used |
| 0 | 1 | 0 | Open | Pull-up | When operating as the function controller (full speed) |
| 0 | 0 | 1 | Pull-up | Open | When operating as the function controller (low speed) |
| 1 | 0 | 0 | Pull-down | Pull-down | When operating as the host controller |
| Other than above | | | — | — | Setting prohibited |

### 25.3.1.4 Example of USB External Connection Circuit

Figure 25.2 shows an example of OTG connection of the USB connector in the self-powered state.
The USB controls the signals for enabling a pull-up resistor for the D+ signal and pull-down resistors for the D+ and D– signals. These signals can be pulled up or down using the SYSCFG.DPRPU and SYSCFG.DRPD bits.
When the function controller is selected and the DPRPU bit is cleared to 0 during communication with the host controller, the pull-up resistor of the USB data line is disabled, making it possible to notify the USB host of the device disconnection.

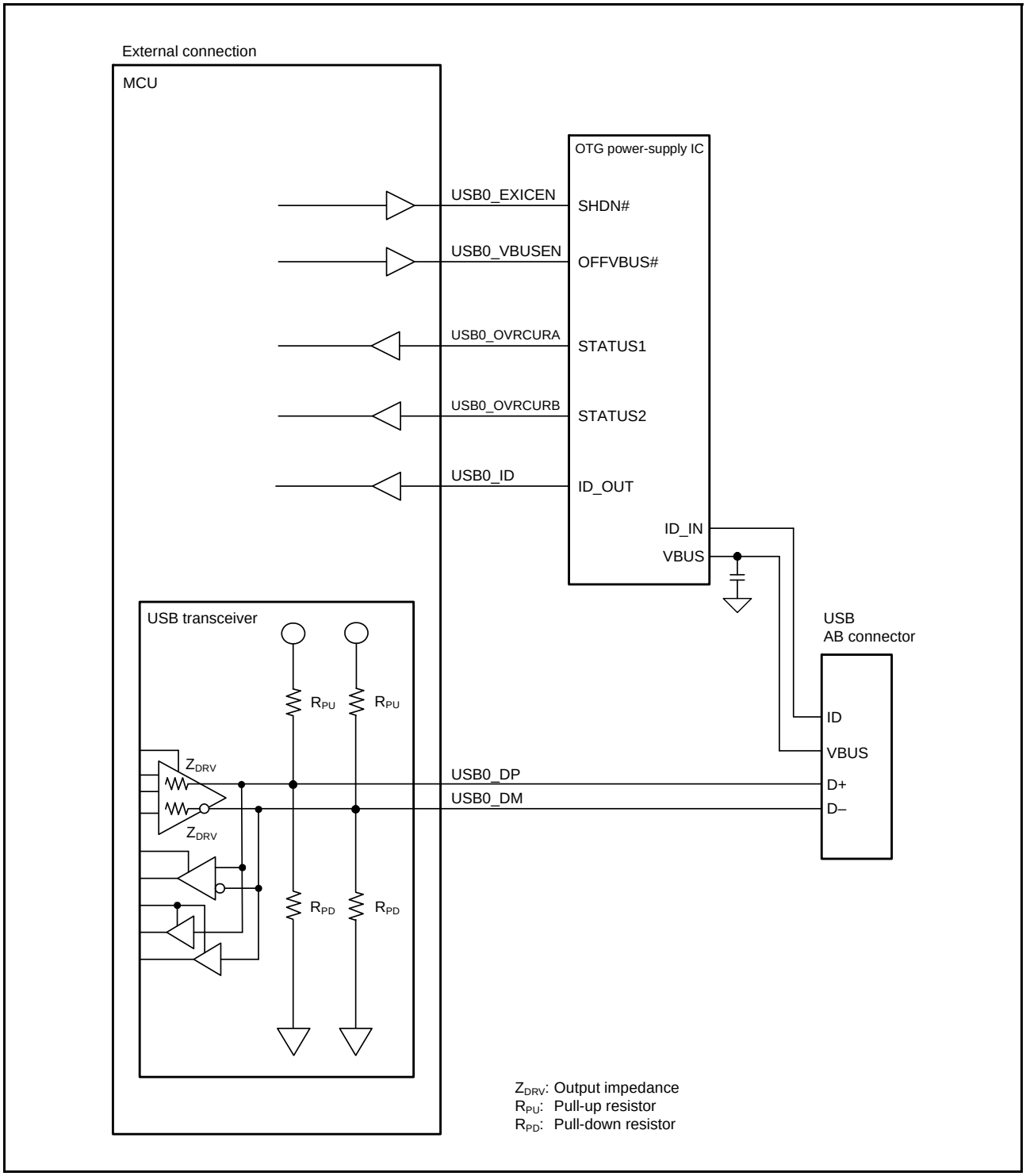



**Figure 25.2 Sample OTG Connection of USB Connector in Self-Powered State**

Figure 25.3 shows an example of functional connection of the USB connector in the self-powered state.

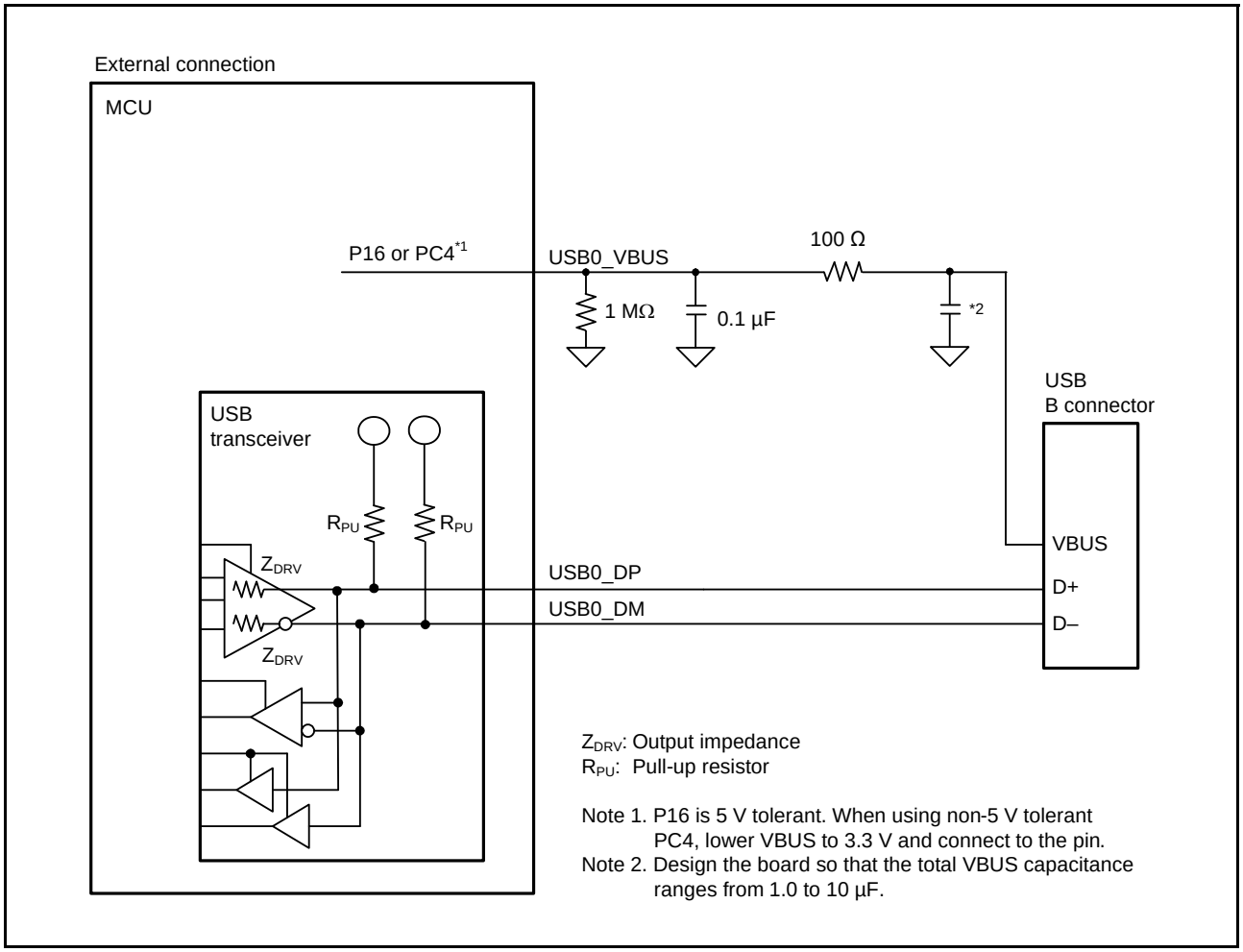



**Figure 25.3 Functional Connection of USB Connector in Self-Powered State**

Figure 25.4 shows an example of functional connection of the USB connector with Battery Charging Rev 1.2 supported.

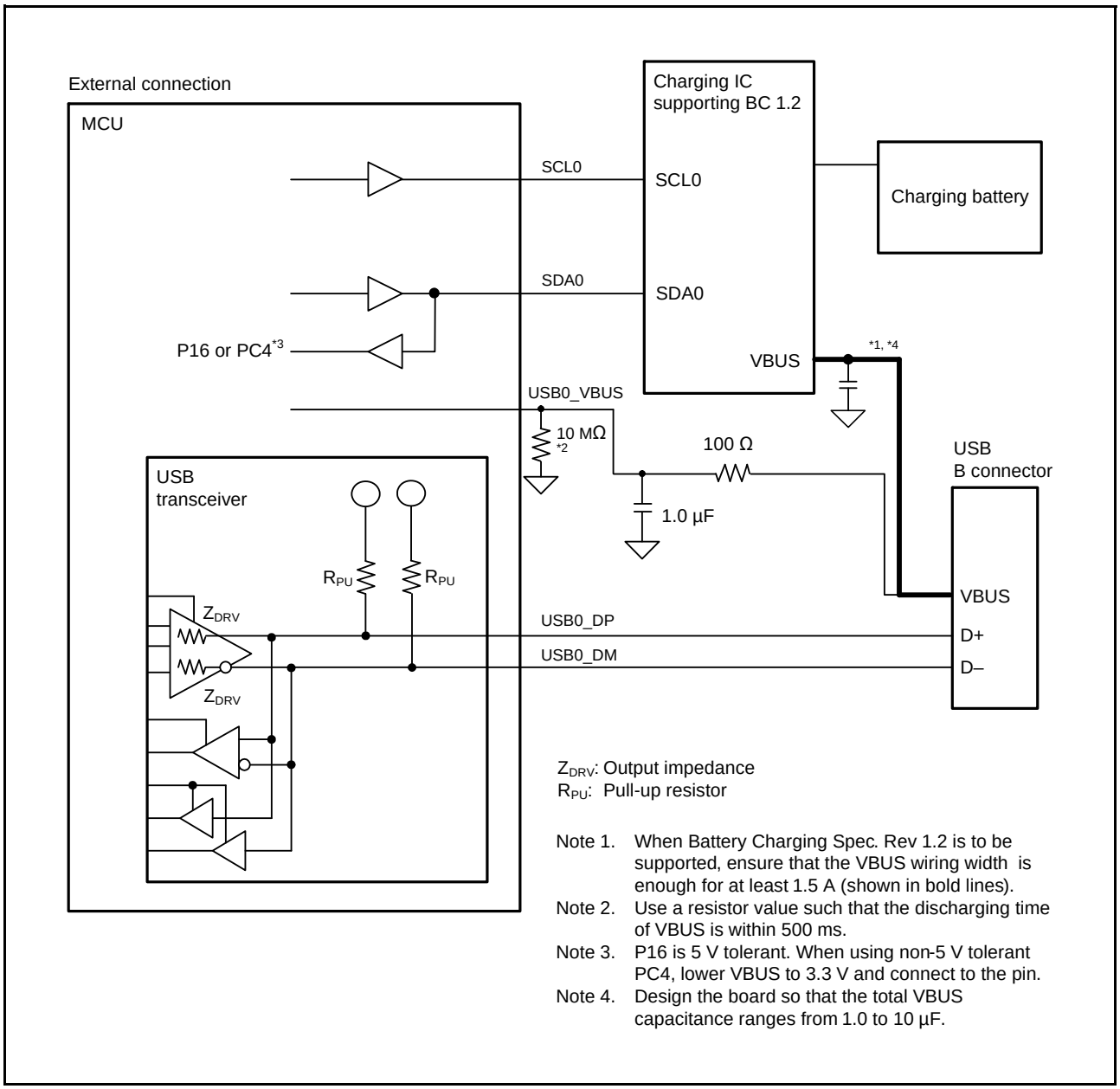



**Figure 25.4 Functional Connection Sample of USB Connector with Battery Charging Rev 1.2 Supported**

Figure 25.5 shows an example of host connection of the USB connector.

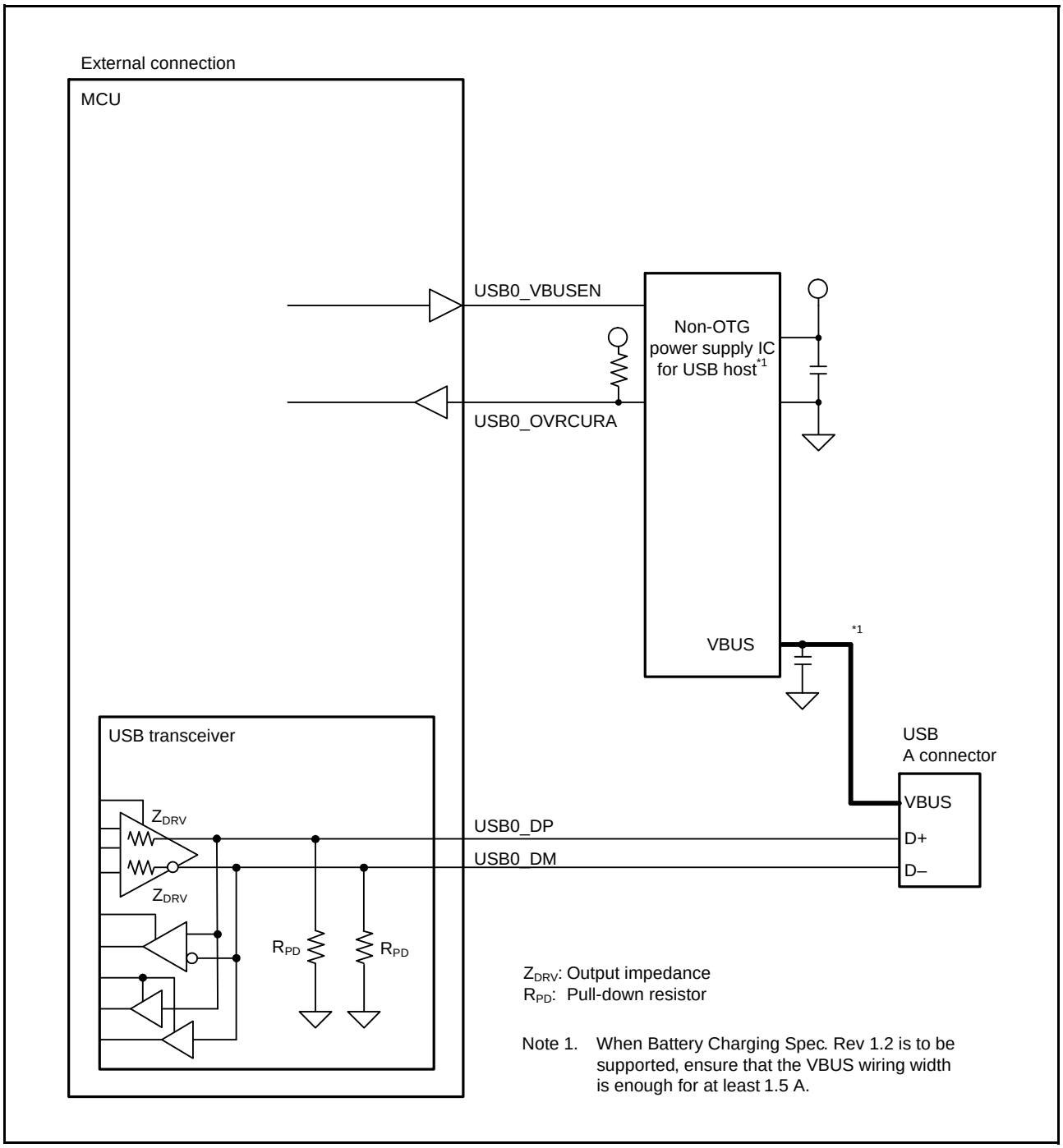



**Figure 25.5 Sample Host Connection of USB Connector**

Figure 25.6 shows an example of functional connection of the USB connector in bus powered state.

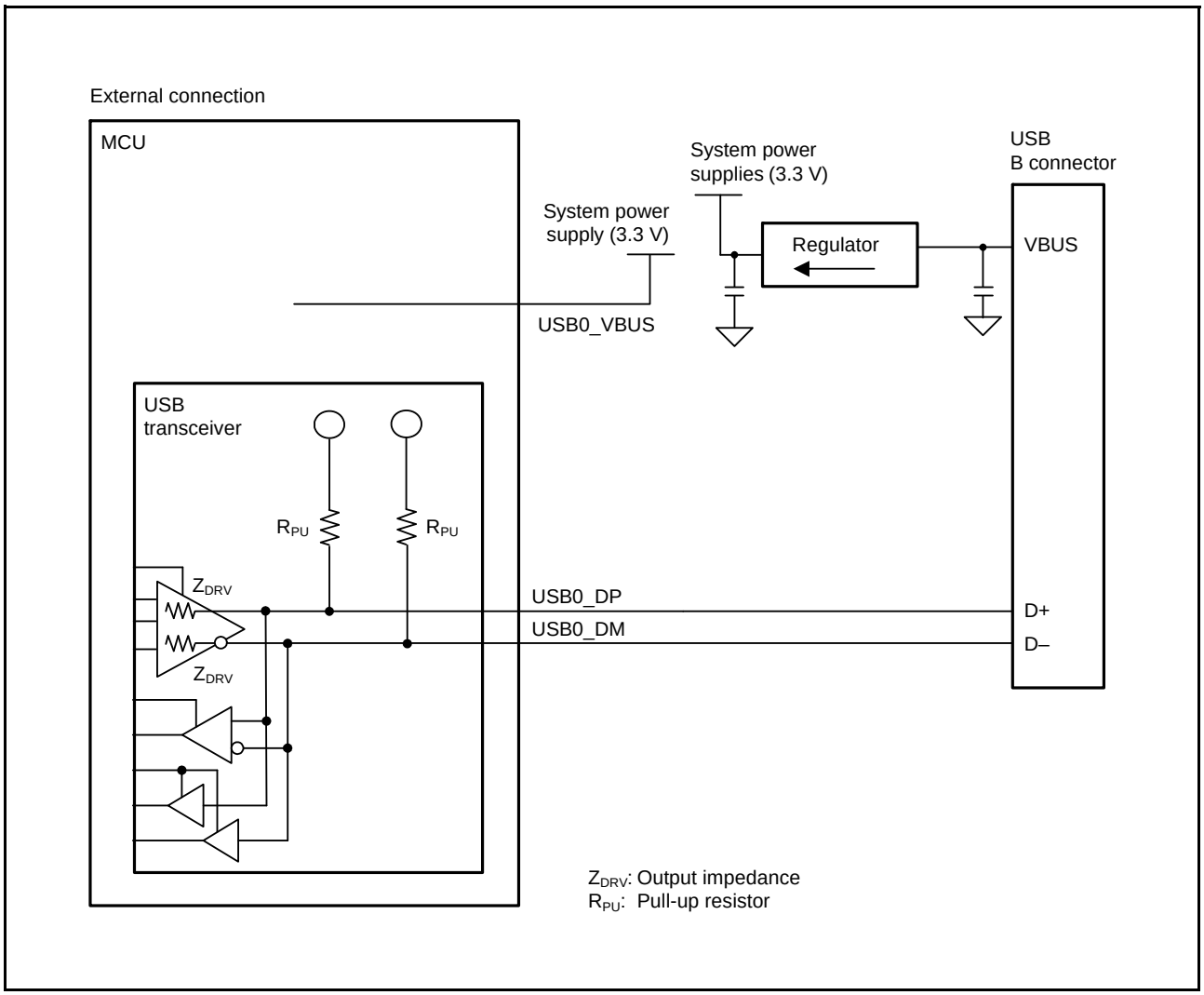



**Figure 25.6 Functional Connection Sample of USB Connector in Bus Powered State**

The examples of external circuits given in this section are simplified circuits, and their operation in every system is not guaranteed.

## 25.3.2 Interrupt Sources

Table 25.13 lists the interrupt sources in the USB.

When an interrupt generation condition is satisfied and the interrupt output is enabled using the corresponding interrupt enable register, a USB interrupt request is issued the Interrupt Controller (ICUb) and an USB interrupt will be generated.

**Table 25.13 Interrupt Sources**

| Bit to be Set | Name | Interrupt Source | Function That Generates the Interrupt | Status Flag |
|---|---|---|---|---|
| VBINT | VBUS interrupt | • When a change in the state of the USBm_VBUS input pin has been detected (low to high or high to low) | Host/function*1 | INTSTS0. VBSTS |
| RESM | Resume interrupt | • When a change in the state of the USB bus has been detected in the suspended state (J-state to K-state or J-state to SE0) | Function | — |
| SOFR | Frame number update interrupt | [Host controller is selected]<br>• When an SOF packet with a different frame number has been transmitted<br>[Function controller is selected]<br>• When an SOF packet with a different frame number has been received | Host/function | — |
| DVST | Device state transition interrupt | • When a device state transition has been detected (any of the following conditions)<br>A USB bus reset detected<br>Suspend state detected<br>SET_ADDRESS request received<br>SET_CONFIGURATION request received | Function | INTSTS0. DVSQ[2:0] |
| CTRT | Control transfer stage transition interrupt | • When a stage transition has been detected in control transfer (any of the following conditions)<br>Setup stage completed<br>Control write transfer status stage transition<br>Control read transfer status stage transition<br>Control transfer completed<br>A control transfer sequence error occurred | Function | INTSTS0. CTSQ[2:0] |
| BEMP | Buffer empty interrupt | • When transmission of all data in the buffer memory has been completed and the buffer has become empty<br>• When a packet larger than the maximum packet size has been received | Host/function | BEMPSTS. PIPEnBEMP |
| NRDY | Buffer not ready interrupt | [Host controller is selected]<br>• When STALL has been received from the peripheral device for the issued token<br>• When a response has not been received correctly from the peripheral device for the issued token (no response was returned three consecutive times or a packet reception error occurred three consecutive times)<br>• When an overrun/underrun occurred during isochronous transfer<br>[Function controller is selected]<br>• When NAK has been returned for an IN or OUT token while the PID bit = BUF<br>• When a CRC error or a bit stuffing error occurred during data reception in isochronous transfer<br>• When an overrun/underrun occurred during data reception in isochronous transfer | Host/function | NRDYSTS. PIPEnNRDY |
| BRDY | Buffer ready interrupt | • When the buffer has become ready (reading or writing is enabled) | Host/function | BRDYSTS. PIPEnBRDY |
| OVRCR | Overcurrent input change interrupt | • When a change in the state of the USB0_OVRCURA or USB0_OVRCURB input pin has been detected (low to high or high to low) | Host | INTSTS1. OVRCR |
| BCHG | Bus change interrupt | • When a change of USB bus state has been detected | Host/function | SYSSTS0. LNST[1:0] |
| DTCH | Disconnection detection during full-speed operation | • When disconnection of a peripheral device has been detected in full-speed operation | Host | DVSTCTR0. RHST[2:0] |
| ATTCH | Device connection detection | • When J-state or K-state is detected on the USB port for 2.5 µs. Used for checking whether a peripheral device is connected. | Host | — |
| EOFERR | EOF error detection | • When an EOF error of a peripheral device has been detected | Host | — |
| SACK | Normal setup operation | • When the normal response (ACK) for the setup transaction has been received | Host | — |
| SIGN | Setup error | • When a setup transaction error (no response or ACK packet corruption) was detected three consecutive times | Host | — |
| PDDETINT0 | Portable device detection interrupt | • When connection of the portable device has been detected | Host | INTSTS1. PDDETINT0 |

Note 1. Though this interrupt can be generated while the host function is selected, it is not usually used with the host function.

Figure 25.7 shows the circuits related to the interrupts in the USB.

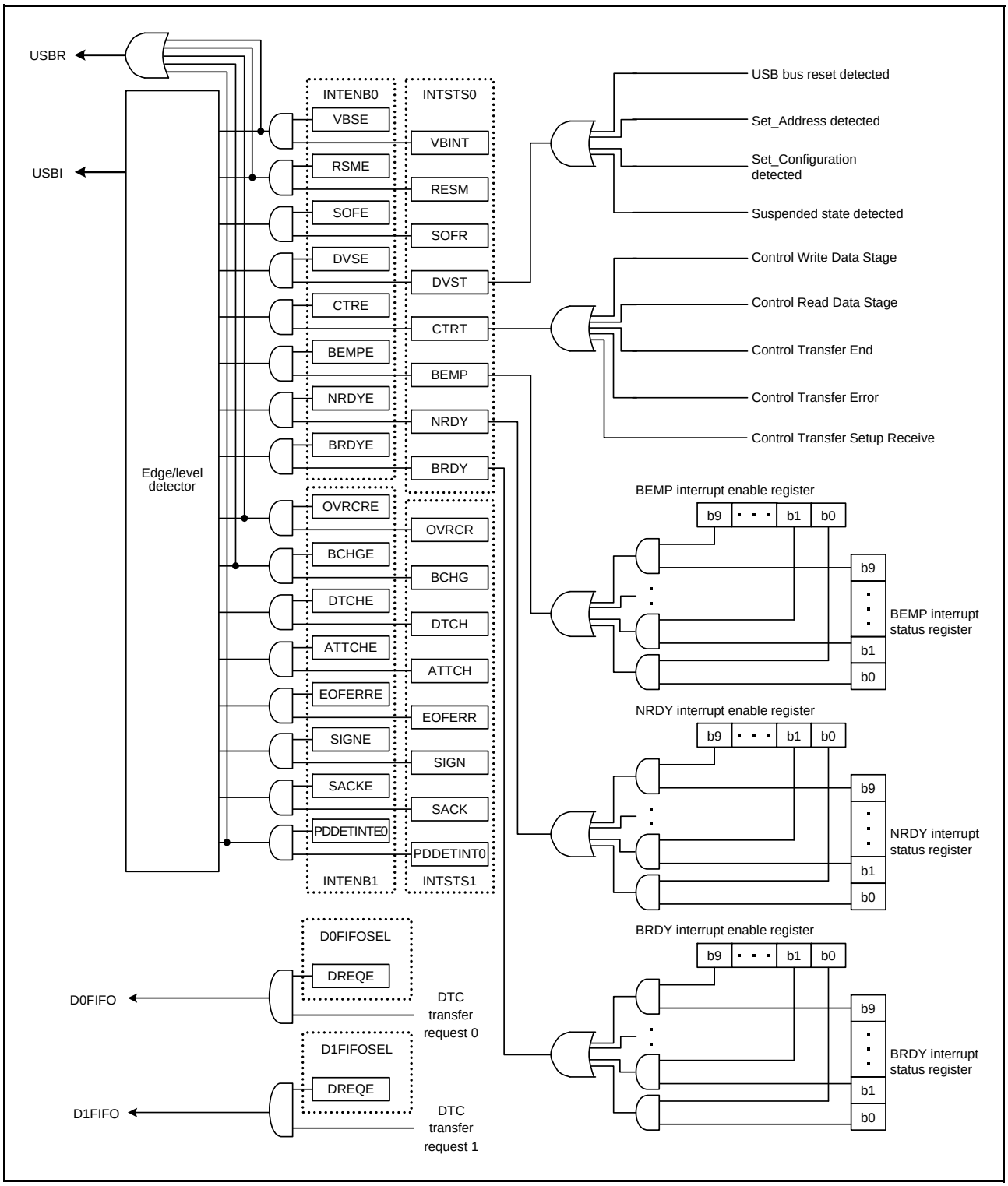



**Figure 25.7 Circuits Related to Interrupts in USB**

Table 25.14 shows the interrupts generated in the USBm (m = 0).

**Table 25.14 USB Interrupts**

| Interrupt Name | Interrupt Status Flag | DTC Activation | Priority |
|---|---|---|---|
| D0FIFO | DTC transfer request 0 | Possible | High |
| D1FIFO | DTC transfer request 1 | Possible | |
| USBI | VBUS interrupt, resume interrupt, frame number update interrupt, device state transition interrupt, control transfer stage transition interrupt, buffer empty interrupt, buffer not ready interrupt, buffer ready interrupt, overcurrent input change interrupt, bus change interrupt, disconnection detection during full-speed operation, device connection detection, EOF error detection, normal setup operation, setup error, and portable device detection interrupt | Not possible | Low |
| USBR | VBUS interrupt, resume interrupt, overcurrent input change interrupt, bus change interrupt, and portable device detection interrupt | Not possible | — |

## 25.3.3 Interrupt Descriptions

### 25.3.3.1 BRDY Interrupt

The BRDY interrupt is generated when either of the host controller or function controller is selected. The following shows the conditions under which the USB sets 1 to a corresponding bit in BRDYSTS. Under this condition, the USB generates a BRDY interrupt if software has set 1 to the BRDYENB.PIPEnBRDYE bit that corresponds to the pipe and 1 to the INTENB0.BRDYE bit.
The conditions for generating and clearing the BRDY interrupt depend on the settings of the SOFCFG.BRDYM bit and PIPECFG.BFRE bit for each pipe as described below.

#### (1) When the SOFCFG.BRDYM Bit = 0 and the PIPECFG.BFRE Bit = 0

With these settings, the BRDY interrupt indicates that the FIFO port is accessible.
On any of the following conditions, the USB generates an internal BRDY interrupt request trigger and sets 1 to the BRDYSTS.PIPEnBRDY bit corresponding to the pertinent pipe.

#### (a) For the pipe in the transmitting direction:

- When the DIR bit is changed from 0 to 1 by software.
- When packet transmission is completed using the pertinent pipe while write-access from the CPU to the FIFO buffer for the pertinent pipe is disabled (when the BSTS bit is read as 0).
- When one FIFO buffer is empty on completion of writing data to the other FIFO buffer in double buffer mode.
- No request trigger is generated until completion of writing data to the currently-written FIFO buffer even if transmission to the other FIFO buffer is completed.
- When the hardware flushes the buffer of the pipe for isochronous transfers.
- When 1 is written to the PIPEnCTR.ACLRM bit, which causes the FIFO buffer to make transition from the write-disabled to write-enabled state.

No request trigger is generated for the DCP (that is, during data transmission for control transfers).

### (b) For the pipe in the receiving direction:

- When packet reception is completed successfully thus enabling the FIFO buffer to be read while read-access from the CPU to the FIFO buffer for the pertinent pipe is disabled (when the BSTS bit is read as 0).
  No request trigger is generated for the transaction in which DATA-PID mismatch has occurred.
- When one FIFO buffer is read-enabled on completion of reading data from the other FIFO buffer in double buffer mode.
  No request trigger is generated until completion of reading data from the currently-read FIFO buffer even if reception by the other FIFO buffer is completed.

When the function controller is selected, the BRDY interrupt is not generated in the status stage of control transfers.
The PIPEBRDY interrupt status of the pertinent pipe can be cleared to 0 by writing 0 to the corresponding PIPEnBRDY bit in BRDYSTS through software. In this case, 1s should be written to the PIPEBRDY bits for the other pipes.
Clear the BRDY status before accessing the FIFO buffer.

## (2) When the SOFCFG.BRDYM Bit = 0 and the PIPECFG.BFRE Bit = 1

With these settings, the USB generates a BRDY interrupt on completion of reading all data for a single transfer using the pipe in the receiving direction, and sets 1 to the bit in BRDYSTS corresponding to the pertinent pipe.
On any of the following conditions, the USB determines that the last data for a single transfer has been received.

- When a short packet including a zero-length packet is received.
- When the transaction counter register (PIPEnTRN) is used and the number of packets specified by the PIPEnTRN.TRNCNT[15:0] bits are completely received.

When the pertinent data is completely read after any of the above conditions has been satisfied, the USB determines that all data for a single transfer has been completely read.
When a zero-length packet is received while the FIFO buffer is empty, the USB determines that all data for a single transfer has been completely read upon passing the zero-length packet data to the CPU. In this case, to start the next transfer, write 1 to the BCLR bit in the corresponding port control register through software.
With these settings, the USB does not detect a BRDY interrupt for the pipe in the transmitting direction.
The PIPEBRDY interrupt status of the pertinent pipe can be cleared to 0 by writing 0 to the corresponding BRDYSTS.PIPEnBRDY bit through software. In this case, 1s should be written to the PIPEBRDY bits for the other pipes.
In this mode, the PIPECFG.BFRE bit setting should not be modified until all data for a single transfer has been processed. When it is necessary to modify the PIPECFG.BFRE bit before completion of processing, all FIFO buffers for the pertinent pipe should be cleared using the PIPEnCTR.ACLRM bit.

### (3) When the SOFCFG.BRDYM Bit = 1 and the PIPECFG.BFRE Bit = 0

With these settings, the BRDYSTS.PIPEnBRDY values are linked to the BSTS bit setting for each pipe. In other words, the BRDY interrupt status bits (PIPEBRDY) are set to 1 or 0 by the USB depending on the FIFO buffer status.

#### (a) For the pipe in the transmitting direction:

The BRDY interrupt status bits are set to 1 when the FIFO buffer is ready for write access, and are set to 0 when it is not ready.
However, the BRDY interrupt is not generated even if the DCP in the transmitting direction is ready for write access.

#### (b) For the pipe in the receiving direction:

The BRDY interrupt status bits are set to 1 when the FIFO buffer is ready for read access, and are set to 0 when all data have been read (not ready for read access).
When a zero-length packet is received while the FIFO buffer is empty, the pertinent bit is set to 1 and the BRDY interrupt is continuously generated until BCLR = 1 is written through software.
With this setting, the PIPEnBRDY bit cannot be cleared to 0 through software.
When the SOFCFG.BRDYM bit is set to 1, all PIPECFG.BFRE bits (for all pipes) should be cleared to 0.

Figure 25.8 shows the timing of BRDY interrupt generation.

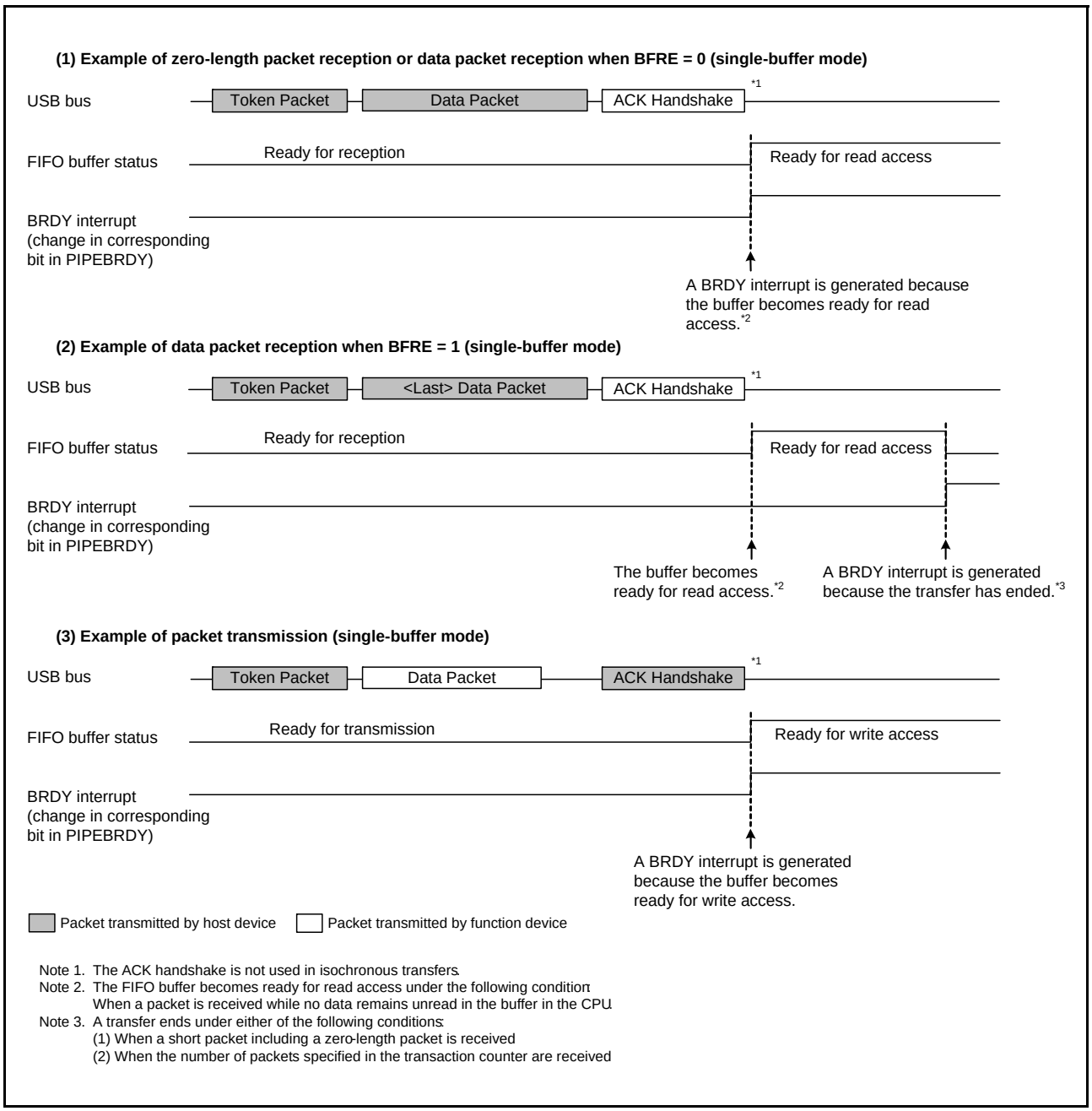



**Figure 25.8 Timing of BRDY Interrupt Generation**

The condition that USB clears the INTSTS0.BRDY bit depends on the SOFCFG.BRDYM bit setting. Table 25.15 shows the condition for clearing the BRDY bit.

**Table 25.15 Condition for Clearing BRDY Bit**

| BRDYM Bit | Condition for Clearing BRDY Bit |
|---|---|
| 0 | The USB clears the BRDY bit when all bits in BRDYSTS have been cleared to 0 by software. |
| 1 | The USB clears the BRDY bit when the BSTS bits for all piles have become 0. |

#### 25.3.3.2 NRDY Interrupt

On generating an internal NRDY interrupt request for the pipe whose PID bits are set to BUF by software, the USB sets the corresponding PIPEnNRDY bit in NRDYSTS to 1. If the corresponding bit in NRDYENB has been set to 1 by software, the USB sets the INTSTS0.NRDY bit to 1 and generates a USB interrupt.
The following describes the conditions on which the USB generates the internal NRDY interrupt request for a given pipe.
Note that the internal NRDY interrupt request is not generated during setup transaction execution when the host controller is selected. During setup transactions when the host controller is selected, the SACK or SIGN interrupt is detected.
The internal NRDY interrupt request is not generated during status stage execution of the control transfer when the function controller is selected.

##### (1) When Host Controller is Selected

##### (a) For the pipe in the transmitting direction:

On any of the following conditions, the USB detects an NRDY interrupt.

- For the pipe for isochronous transfers, when the time to issue an OUT token comes while there is no data to be transmitted in the FIFO buffer.
  In this case, the USB transmits a zero-length packet following the OUT token and sets the bit corresponding to the NRDYSTS.PIPEnNRDY bit and the FRMNUM.OVRN bit to 1.
- During communications other than setup transactions using the pipe for the transfers other than isochronous transfers, when any combination of the following two cases occur three consecutive times: 1) no response is returned from the peripheral device (when timeout is detected before detection of the handshake packet from the peripheral device) and 2) an error is detected in the packet from the peripheral device.
  In this case, the USB sets the bit corresponding to the PIPEnNRDY bit to 1 and modifies the setting of the PID[1:0] bits of the corresponding pipe to NAK.
- During communications other than setup transactions, when the STALL handshake is received from the peripheral device.
  In this case, the USB sets the bit corresponding to the PIPEnNRDY bit to 1 and modifies the setting of the PID[1:0] bits of the corresponding pipe to STALL (11b).

##### (b) For the pipe in the receiving direction:

- For the pipe for isochronous transfers, when the time to issue an IN token comes while there is no space available in the FIFO buffer.
  In this case, the USB discards the received data for the IN token and sets the PIPEnNRDY bit corresponding to the pipe and the OVRN bit to 1.
  When a packet error is detected in the received data for the IN token, the USB also sets the FRMNUM.CRCE bit to 1.
- For the pipe for the transfers other than isochronous transfers, when any combination of the following two cases occur three consecutive times: 1) no response is returned from the peripheral device for the IN token issued by the USB (when timeout is detected before detection of the DATA packet from the peripheral device) and 2) an error is detected in the packet from the peripheral device.
  In this case, the USB sets the PIPEnNRDY bit corresponding to the pipe to 1 and modifies the setting of the PID[1:0] bits of the corresponding pipe to NAK.
- For the pipe for isochronous transfers, when no response is returned from the peripheral device for the IN token (when timeout is detected before detection of the DATA packet from the peripheral device) or an error is detected in the packet from the peripheral device.
  In this case, the USB sets the PIPEnNRDY bit corresponding to the pipe to 1. (The setting of the PID[1:0] bits of the pipe is not modified.)

- For the pipe for isochronous transfers, when a CRC error or a bit stuffing error is detected in the received data packet.
  In this case, the USB sets the PIPEnNRDY bit corresponding to the pipe and the CRCE bit to 1.
- When the STALL handshake is received.
  In this case, the USB sets the PIPEnNRDY bit corresponding to the pipe to 1 and modifies the setting of the PID[1:0] bits of the corresponding pipe to STALL.

### (2) When Function Controller is Selected

#### (a) For the pipe in the transmitting direction:

- When an IN token is received while there is no data to be transmitted in the FIFO buffer.
  In this case, the USB generates a NRDY interrupt request at the reception of the IN token and sets the NRDYSTS.PIPEnNRDY bit to 1.
  For the pipe for the isochronous transfers in which an interrupt is generated, the USB transmits a zero-length packet and sets the FRMNUM.OVRN bit to 1.

#### (b) For the pipe in the receiving direction:

- When an OUT token is received while there is no space available in the FIFO buffer.
  For the pipe for the isochronous transfers in which an interrupt is generated, the USB generates a NRDY interrupt request at the reception of the OUT token and sets the PIPEnNRDY bit to 1 and OVRN bit to 1.
  For the pipe for the transfers other than isochronous transfers in which an interrupt is generated, the USB generates a NRDY interrupt request when a NAK handshake is transferred after the data following the OUT token is received, and sets the PIPEnNRDY bit to 1.
  However, during re-transmission (due to DATA-PID mismatch), the NRDY interrupt request is not generated. In addition, if an error occurs in the DATA packet, the NRDY interrupt request is not generated.
- For the pipe for isochronous transfers, when a token is not received successfully within an interval frame.
  In this case, the USB generates a NRDY interrupt request when SOF is received, and sets the PIPEnNRDY bit to 1.

Figure 25.9 shows the timing of NRDY interrupt generation when the function controller is selected.

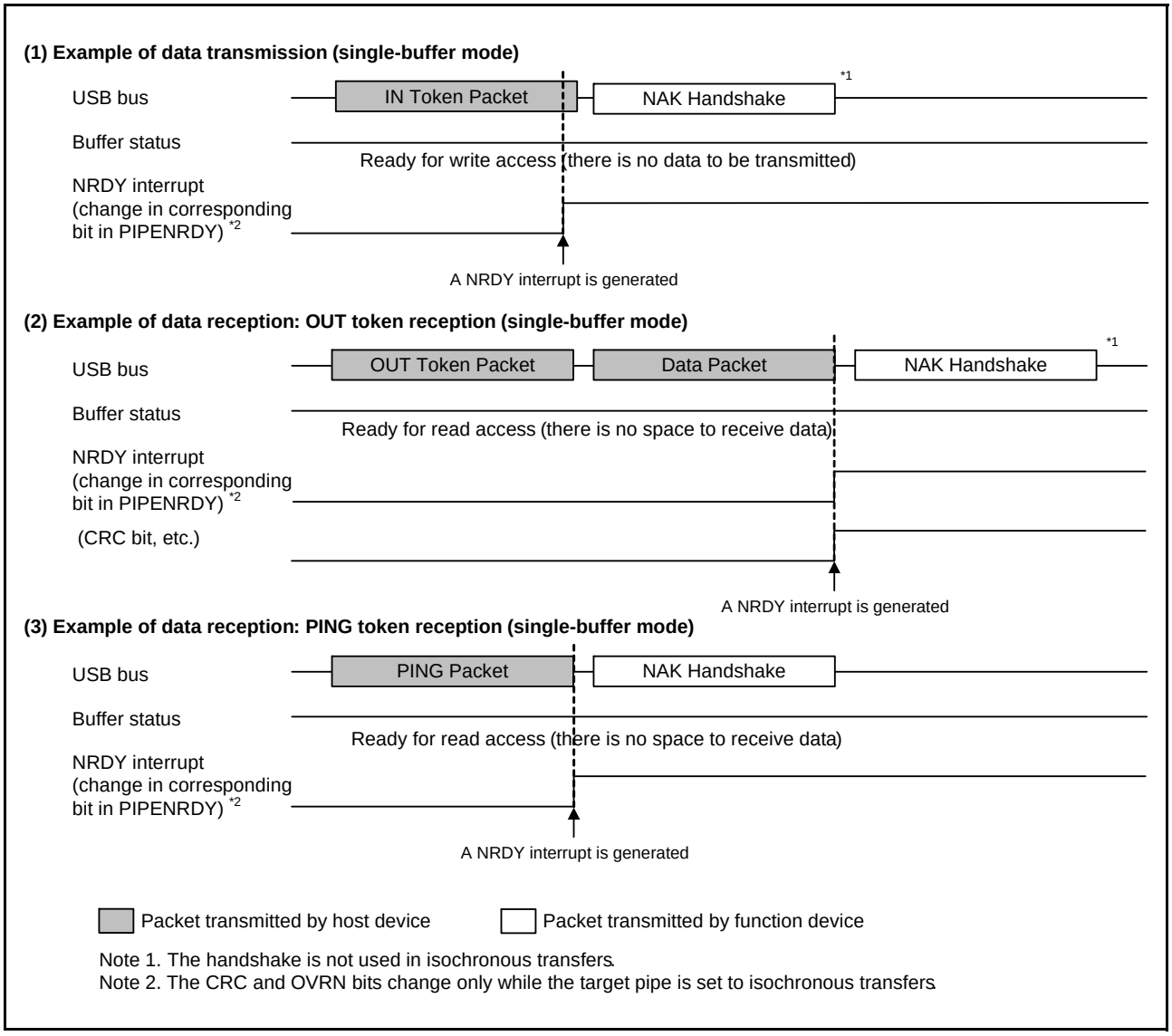



**Figure 25.9 Timing of NRDY Interrupt Generation (When Function Controller is Selected)**

### 25.3.3.3 BEMP Interrupt

On detecting a BEMP interrupt for the pipe whose PID bits are set to BUF by software, the USB sets the corresponding BEMPSTS.PIPEnBEMP bit to 1. If the corresponding bit in BEMPENB has been set to 1 by software, the USB sets the INTSTS0.BEMP bit to 1 and generates a USB interrupt.

The following describes the conditions on which the USB generates an internal BEMP interrupt request.

#### (1) For the pipe in the transmitting direction:

When the FIFO buffer of the corresponding pipe is empty on completion of transmission (including zero-length packet transmission).
In single buffer mode, an internal BEMP interrupt request is generated simultaneously with the BRDY interrupt for the pipe other than DCP. However, the internal BEMP interrupt request is not generated on any of the following conditions.

- When the CPU or DTC has already started writing data to the FIFO buffer of the CPU on completion of transmitting data from one FIFO buffer in double buffer mode.
- When the buffer is cleared (emptied) by setting the PIPEnCTR.ACLRM or the BCLR bit in the port control register to 1.
- When IN transfer (zero-length packet transmission) is performed during the control transfer status stage while the function controller is selected.

#### (2) For the pipe in the receiving direction:

When the successfully-received data packet size exceeds the specified maximum packet size.
In this case, the USB generates a BEMP interrupt request, sets the corresponding BEMPSTS.PIPEnBEMP bit to 1, discards the received data, and modifies the setting of the PID[1:0] bits of the corresponding pipe to STALL (11b). Here, the USB returns no response when used as the host controller, and returns STALL response when used as the function controller.
However, the internal BEMP interrupt request is not generated on any of the following conditions.

- When a CRC error or a bit stuffing error is detected in the received data.
- When a setup transaction is being performed,
  Writing 0 to the BEMPSTS.PIPEnBEMP bit clears the status.
  Writing 1 to the BEMPSTS.PIPEnBEMP bit has no effect.

Figure 25.10 shows the timing of BEMP interrupt generation when the function controller is selected.

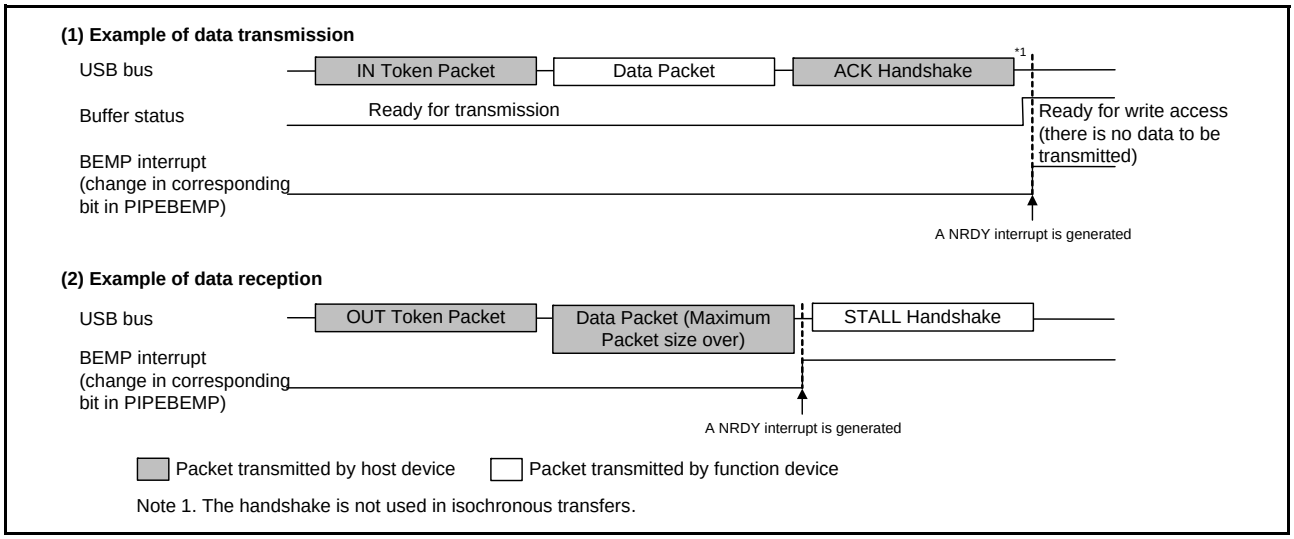



**Figure 25.10 Timing of BEMP Interrupt Generation (When Function Controller is Selected)**

### 25.3.3.4 Device State Transition Interrupt

Figure 25.11 is a diagram of device state transitions in the USB. The USB controls device state and generates device state transition interrupts. However, recovery from the suspended state (resume signal detection) is detected by means of the resume interrupt. The device state transition interrupts can be enabled or disabled individually using INTENB0. The device state to which a transition was made can be confirmed using the DVSQ bits in INTSTS0.

When a transition is made to the default state, a device state transition interrupt is generated after a USB bus reset is detected.

Device state can be controlled only when the function controller is selected. The device state transition interrupts can also be generated only when the function controller is selected.

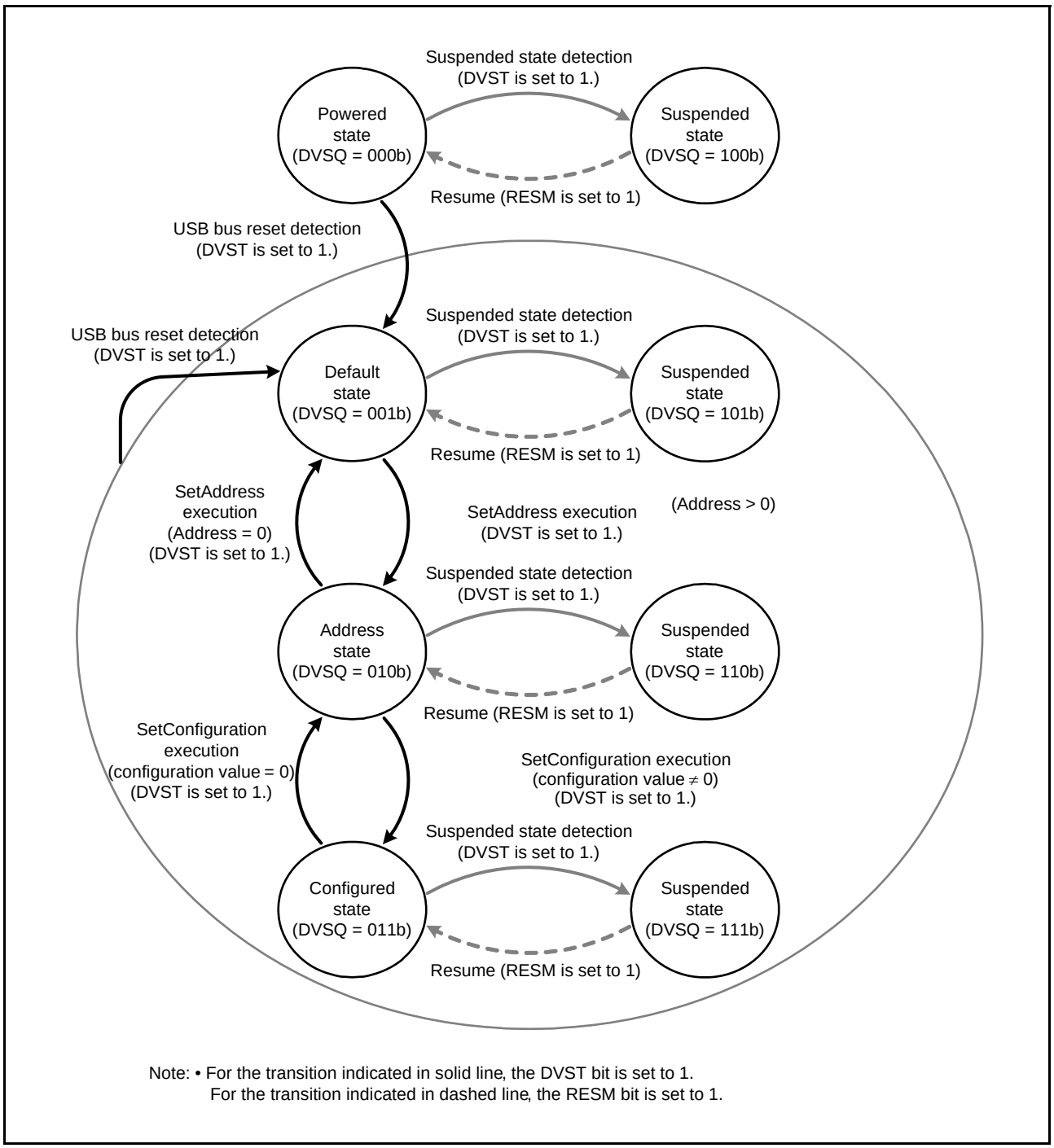



**Figure 25.11 Device State Transitions**

### 25.3.3.5 Control Transfer Stage Transition Interrupt

Figure 25.12 is a diagram of control transfer stage transitions in the USB. The USB controls the control transfer sequence and generates control transfer stage transition interrupts. The control transfer stage transition interrupts can be enabled or disabled individually using INTENB0. The transfer stage to which a transition was made can be confirmed using the CTSQ[2:0] bits in INTSTS0.

Control transfer stage transition interrupts are generated only when the function controller is selected.

The control transfer sequence errors are listed below. If an error occurs, the DCPCTR.PID[1:0] bits are set to 1xb (STALL response).

During control read transfer:

- An OUT token is received while no data has been transferred for the IN token at the data stage.
- An IN token is received at the status stage.
- A data packet with DATAPID = DATA0 is received at the status stage.

During control write transfer:

- An IN token is received while no ACK response has been returned for the OUT token at the data stage.
- A data packet with DATAPID = DATA0 is received for the first data packet at the data stage.
- An OUT token is received at the status stage

During no-data control transfers:

- An OUT token is received at the status stage.

At the control write transfer data stage, if the number of receive data exceeds the wLength value of the USB request, it cannot be recognized as a control transfer sequence error. At the control read transfer status stage, packets other than zero-length packets are received by an ACK response and the transfer ends normally.

When a CTRT interrupt occurs in response to a sequence error (INTSTS0.CTRT = 1), CTSQ[2:0] = 110b value is retained until the CTRT bit = 0 is written from the system (the interrupt status is cleared). Therefore, while CTSQ[2:0] = 110b is being held, the CTRT interrupt that ends the setup stage will not be generated even if a new USB request is received. (The USB retains the setup stage end, and after the interrupt status has been cleared by software, a setup stage end interrupt is generated.)

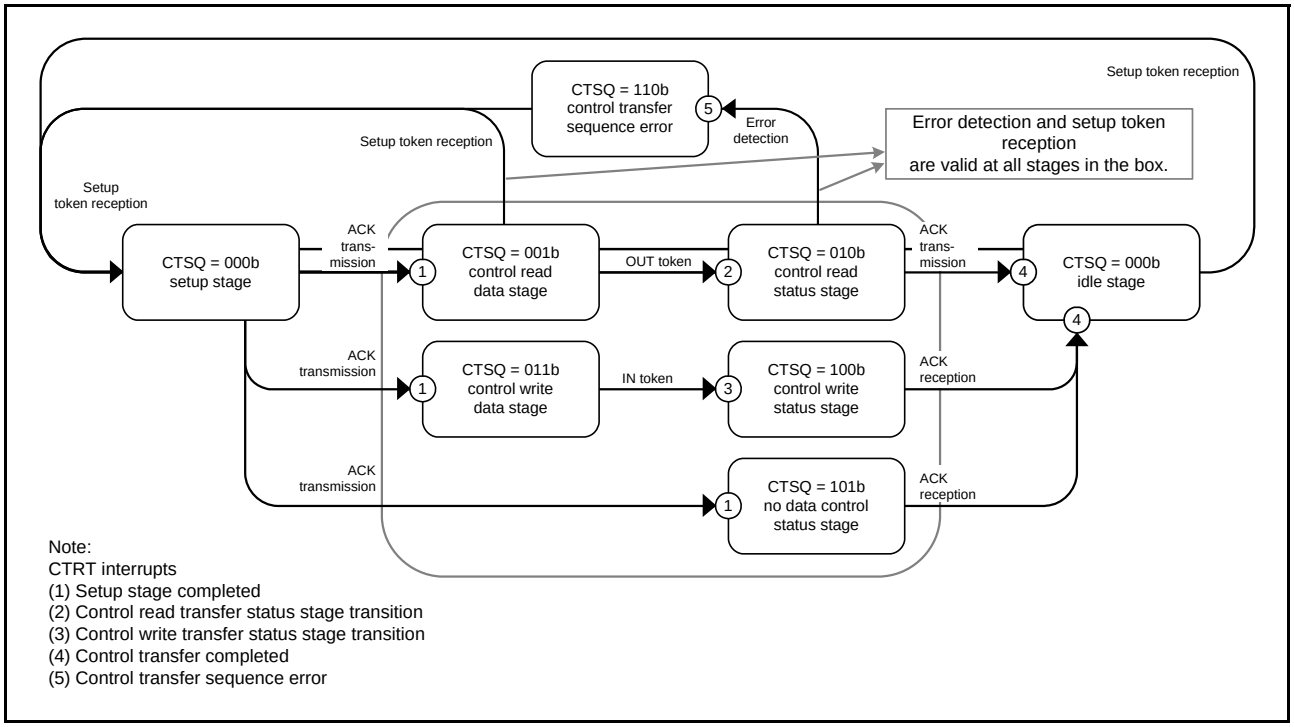



**Figure 25.12 Control Transfer Stage Transitions**

#### 25.3.3.6 Frame Update Interrupt

With the host controller selected, an interrupt is generated at the timing when the frame number is updated. With the function controller selected, an SOFR interrupt is generated when the frame number is updated.
When the function controller is selected, the USB updates the frame number and generates an SOFR interrupt if it detects a new SOF packet during full-speed operation.

#### 25.3.3.7 VBUS Interrupt

When the USBm_VBUS pin level changes, a VBUS interrupt is generated. The level of the USBm_VBUS pin can be checked with the INTSTS0.VBSTS bit. Whether the host controller is connected or disconnected can be confirmed using the VBUS interrupt. However, if the system is activated with the host controller connected, the first VBUS interrupt is not generated because there is no change in the USBm_VBUS pin level.

#### 25.3.3.8 Resume Interrupt

When the function controller is selected, a resume interrupt is generated when the device state is the suspended state and the USB bus state has changed (from J-state to K-state, or from J-state to SE0). Recovery from the suspended state is detected by means of the resume interrupt.
When the host controller is selected, no resume interrupt is generated. Use the BCHG interrupt to detect a change in the USB bus state.

#### 25.3.3.9 OVRCR Interrupt

An OVRCR interrupt is generated when the USB0_OVRCURA or USB0_OVRCURB pin level has changed. The levels of the USB0_OVRCURA and USB0_OVRCURB pins can be checked with the SYSSTS0.OVCMON[1:0] bits. The external power-supply IC can check whether overcurrent has been detected using the OVRCR interrupt.
For OTG connection, whether a change has been detected in the VBUS comparator can be checked using the OVRCR interrupt.

#### 25.3.3.10 BCHG Interrupt

A BCHG interrupt is generated when the USB bus state has changed. The BCHG interrupt can be used to detect whether the peripheral device is connected and can also be used to detect a remote wakeup when the host controller is selected. The BCHG interrupt is generated regardless of whether the host controller or function controller is selected.

#### 25.3.3.11 DTCH Interrupt

A DTCH interrupt is generated when disconnection of the USB bus is detected while the host controller is selected. The USB detects bus disconnection based on USB Specifications 2.0.
After detecting a DTCH interrupt, the USB controls hardware as described below (irrespective of the value set in the corresponding interrupt enable bit). All pipes in which communications are currently carried out for the pertinent port should be terminated by software and make a transition to the wait state for bus connection to the pertinent port (wait state for ATTCH interrupt generation).

- Modifies the DVSTCTR0.UACT bit for the port in which a DTCH interrupt has been detected to 0.
- Puts the port in which a DTCH interrupt has been generated into the idle state.

#### 25.3.3.12 SACK Interrupt

A SACK interrupt is generated when an ACK response for the transmitted setup packet has been received from the peripheral device with the host controller selected. The SACK interrupt can be used to confirm that the setup transaction has been completed successfully.

#### 25.3.3.13 SIGN Interrupt

A SIGN interrupt is generated when an ACK response for the transmitted setup packet has not been correctly received from the peripheral device three consecutive times with the host controller selected. The SIGN interrupt can be used to detect no ACK response transmitted from the peripheral device or corruption of an ACK packet.

#### 25.3.3.14 ATTCH Interrupt

An ATTCH interrupt is generated when J-state or K-state of the full-speed signal level is detected on the USB port for 2.5 µs with the host controller selected. To be more specific, an ATTCH interrupt is detected on any of the following conditions.

- When K-state, SE0, or SE1 changes to J-state, and J-state continues 2.5 µs.
- When J-state, SE0, or SE1 changes to K-state, and K-state continues 2.5 µs.

#### 25.3.3.15 EOFERR Interrupt

An EOFERR interrupt is generated when it is detected that communication is not completed at the EOF2 timing prescribed in USB Specifications 2.0.
After detecting an EOFERR interrupt, the USB controls hardware as described below (irrespective of the value set in the corresponding interrupt enable bit). All pipes in which communications are currently carried out for the pertinent port should be terminated by software and perform re-enumeration of the pertinent port.

- Modifies the DVSTCTR0.UACT bit for the port in which an EOFERR interrupt has been detected to 0 (n = 0, 1).
- Puts the port in which an EOFERR interrupt has been generated into the idle state.

#### 25.3.3.16 Portable Device Detection Interrupt

A portable device detection interrupt is generated when the USB module detects a level change (high to low or low to high) in the PDDET output from the USB-PHY. When a portable device detection interrupt is generated, use software to repeat reading the PDDETSTS bit until the same value is read three or more times, and perform debouncing.

### 25.3.4 Pipe Control

Table 25.16 lists the pipe settings for the USB. With USB data transfer, data transfer is carried out using the pipe that the software has associated with the endpoint. The USB has ten pipes that are used for data transfer.
Appropriate settings should be made for each of the pipes according to the specifications of the system.

**Table 25.16 Pipe Settings**

| Register Name | Bit Name | Setting | Remarks |
|---|---|---|---|
| DCPCFG<br>PIPECFG | TYPE | Specifies the transfer type | PIPE1 to PIPE9: Can be set |
| | BFRE | Selects the BRDY interrupt mode | PIPE1 to PIPE5: Can be set |
| | DBLB | Selects double buffer mode | PIPE1 to PIPE5: Can be set |
| | DIR | Selects transfer direction | IN or OUT can be set |
| | EPNUM | Endpoint number | PIPE1 to PIPE9: Can be set<br>A value other than 0000b should be set when the pipe is used. |
| | SHTNAK | Selects disabled state for pipe when transfer ends | PIPE1 and PIPE2: Can be set (only when bulk transfer has been selected)<br>PIPE3 to PIPE5: Can be set |
| DCPMAXP<br>PIPEMAXP | DEVSEL | Selects a device | Referenced only when the host controller is selected. |
| | MXPS | Maximum packet size | Compliant with USB Specification 2.0. |
| PIPEPERI | IFIS | Buffer flush | PIPE1 and PIPE2: Can be set (only when isochronous transfer has been selected)<br>PIPE3 to PIPE9: Cannot be set |
| | IITV | Interval counter | PIPE1 and PIPE2: Can be set (only when isochronous transfer has been selected)<br>PIPE3 to PIPE5: Cannot be set<br>PIPE6 to PIPE9: Can be set (only when the host controller has been selected) |
| DCPCTR<br>PIPEnCTR | BSTS | Buffer status | For the DCP, receive buffer status and transmit buffer status are switched with the ISEL bit. |
| | INBUFM | IN buffer monitor | Available only for PIPE1 to PIPE5. |
| | SUREQ | SETUP request | Can be set only for the DCP.<br>Can be controlled only when the host controller has been selected. |
| | SUREQCLR | SUREQ clear | Can be set only for the DCP.<br>Can be controlled only when the host controller has been selected. |
| | ATREPM | Auto response mode | PIPE1 to PIPE5: Can be set<br>Can be set only when the function controller has been selected. |
| | ACLRM | Auto buffer clear | PIPE1 to PIPE9: Can be set |
| | SQCLR | Sequence clear | Clears the data toggle bit. |
| | SQSET | Sequence set | Sets the data toggle bit. |
| | SQMON | Sequence monitor | Monitors the data toggle bit. |
| | PBUSY | Pipe busy status | |
| | PID | Response PID | Refer to section 25.3.4.6, Response PID. |
| PIPEnTRE | TRENB | Transaction counter enable | PIPE1 to PIPE5: Can be set |
| | TRCLR | Current transaction counter clear | PIPE1 to PIPE5: Can be set |
| PIPEnTRN | TRNCNT | Transaction counter | PIPE1 to PIPE5: Can be set |

#### 25.3.4.1 Pipe Control Register Switching Procedures

The following bits in the pipe control registers can be modified only when USB communication is prohibited (PID = NAK).

The following shows the registers and bits that should not be modified when USB communication is enabled (PID = BUF).

- Bits in DCPCFG and DCPMAXP
- SQCLR and SQSET bits in DCPCTR
- Bits in PIPECFG, PIPEMAXP, and PIPEPERI
- ATREPM, ACLRM, SQCLR, and SQSET bits in PIPEnCTR
- Bits in PIPEnTRE and PIPEnTRN

In order to modify the above bits in the USB communication enabled (PID = BUF) state, follow the procedure shown below:

1. A request to modify bits in the pipe control register occurs.
2. Modify the PID[1:0] bit corresponding to the pipe to NAK.
3. Wait until the corresponding PBUSY bit is cleared to 0.
4. Modify the bits in the pipe control register.

The following bits in the pipe control registers can be modified only when the pertinent pipe information has not been set by the CURPIPE[3:0] bits in CFIFOSEL, D0FIFOSEL, and D1FIFOSEL.

Registers that should not be set when the CURPIPE[3:0] bits are set:

- Bits in DCPCFG and DCPMAXP
- Bits in PIPECFG, PIPEMAXP and PIPEPERI

In order to modify pipe information, the CURPIPE[3:0] bits in the port select registers should be set to a pipe other than the pipe to be modified. For the DCP, the buffer should be cleared using the BCLR bit in the port control register after the pipe information is modified.

#### 25.3.4.2 Transfer Types

The PIPECFG.TYPE[1:0] bits are used to specify the transfer type for each pipe. The transfer types that can be set for the pipes are as follows.

- DCP: No setting is necessary (fixed at control transfer).
- PIPE1 and PIPE2: These should be set to bulk transfer or isochronous transfer.
- PIPE3 to PIPE5: These should be set to bulk transfer.
- PIPE6 to PIPE9: These should be set to interrupt transfer.

#### 25.3.4.3 Endpoint Number

The PIPECFG.EPNUM[3:0] bits are used to set the endpoint number for each pipe. The DCP is fixed at endpoint 0. The other pipes can be set from endpoint 1 to endpoint 15.

- DCP: No setting is necessary (fixed at endpoint 0).
- PIPE1 to PIPE9: The endpoint numbers from 1 to 15 should be selected and set.
  These should be set so that the combination of the PIPECFG.DIR bit and EPNUM[3:0] bits is unique.

#### 25.3.4.4 Maximum Packet Size Setting

The DCPMAXP.MXPS[6:0] bits and the PIPEMAXP.MXPS[8:0] bits are used to specify the maximum packet size for each pipe. DCP and PIPE1 to PIPE5 can be set to any of the maximum pipe sizes defined by USB Specifications 2.0. For PIPE6 to PIPE9, 64 bytes are the upper limit of the maximum packet size. The maximum packet size should be set before beginning the transfer (PID = BUF).

- DCP: Set 8, 16, 32, or 64.
- PIPE1 to PIPE5: Set 8, 16, 32, or 64 when using bulk transfer.
- PIPE1 and PIPE2: Set a value between 1 and 256 when using isochronous transfer.
- PIPE6 to PIPE9: Set a value between 1 and 64.

#### 25.3.4.5 Transaction Counter (For PIPE1 to PIPE5 in Reading Direction)

When the specified number of transactions has been completed in the data packet receiving direction, the USB recognizes that the transfer has ended. Two transaction counters are provided: one is the PIPEnTRN register that specifies the number of transactions to be executed and the other is the current counter that internally counts the number of executed transactions. With the PIPECFG.SHTNAK bit set to 1, when the current counter value matches the specified number of transactions, the corresponding PIPEnCTR.PID[1:0] bits are set to NAK and the subsequent transfer is disabled. The transactions can be counted again from the beginning by initializing the current counter of the transaction counter function through the PIPEnTRE.TRCLR bit. The information read from PIPEnTRN differs depending on the setting of the PIPEnTRE.TRENB bit.

- The TRENB bit = 0: The specified transaction counter value can be read.
- The TRENB bit = 1: The current counter value indicating the internally counted number of executed transactions can be read.

When operating the TRCLR bit, the following should be noted.

- If the transactions are being counted and PID = BUF, the current counter cannot be cleared.
- If there is any data left in the buffer, the current counter cannot be cleared.

#### 25.3.4.6 Response PID

The PID[1:0] bits in DCPCTR and PIPEnCTR are used to set the response PID for each pipe.
The following shows the USB operation with various response PID settings:

##### (1) Response PID settings when the host controller is selected:

The response PID is used to specify the execution of transactions.

- NAK setting: Using pipes is disabled. No transaction is executed.
- BUF setting: Transactions are executed based on the status of the buffer memory.
  For OUT direction: If there are transmit data in the buffer memory, an OUT token is issued.
  For IN direction: If there is an area to receive data in the buffer memory, an IN token is issued.
- STALL setting: Using pipes is disabled. No transaction is executed.

Note: • Setup transactions for the DCP are set with the DCPCTR.SUREQ bit.

##### (2) Response PID settings when the function controller is selected:

The response PID is used to specify the response to transactions from the host.

- NAK setting: The NAK response is returned in response to the generated transaction.
- BUF setting: Responses are made to transactions according to the status of the buffer memory.
- STALL setting: The STALL response is returned in response to the generated transaction.

Note: • For setup transactions, an ACK response is returned regardless of the PID[1:0] bits setting, and the USB request is stored in the register.

The USB may write to the PID[1:0] bits, depending on the results of the transaction as described below.

##### (3) When the host controller has been selected and the response PID is set by hardware:

- NAK setting: In the following cases, PID = NAK is set and issuing of tokens is automatically stopped:
  When a transfer other than isochronous transfer has been performed and an NRDY interrupt is generated. (For details, refer to section 25.3.3.2, NRDY Interrupt.)
  - If a short packet is received when the PIPECFG.SHTNAK bit has been set to 1 for bulk transfer.
  - If the transaction counting ends when the SHTNAK bit has been set to 1 for bulk transfer.
- BUF setting: There is no BUF writing by the USB.
- STALL setting: In the following cases, PID = STALL is set and issuing of tokens is automatically stopped:
  When STALL is received in response to the transmitted token.
  When the size of the receive data packet exceeds the maximum packet size.

##### (4) When the function controller has been selected and the response PID is set by hardware:

- NAK setting: In the following cases, PID = NAK is set and NAK is returned in response to transactions:
  When the SETUP token is received normally (DCP only).
  If the transaction counting ends or a short packet is received when the PIPECFG.SHTNAK bit has been set to 1 for bulk transfer.
- BUF setting: There is no BUF writing by the USB.
- STALL setting: In the following cases, PID = STALL is set and STALL is returned in response to transactions:
  When a maximum packet size exceeded error is detected in the received data packet.
  When a control transfer sequence error has been detected (DCP only).

#### 25.3.4.7 Data PID Sequence Bit

The USB automatically toggles the sequence bit in the data PID when data is transferred successfully in the control transfer data stage, bulk transfer, and interrupt transfer. The sequence bit of the next data PID to be transmitted can be confirmed with the SQMON bit in DCPCTR and PIPEnCTR. When data is transmitted, the sequence bit switches at the timing of ACK handshake reception. When data is received, the sequence bit switches at the timing of ACK handshake transmission. The SQCLR bit in DCPCTR and the SQSET bit in PIPEnCTR can be used to change the data PID sequence bit.

When the function controller has been selected and control transfer is used, the USB automatically sets the sequence bit when a stage transition is made. DATA1 is returned when the setup stage is ended. The sequence bit is not referenced and PID = DATA1 is returned in a status stage. Therefore, software settings are not required. However, when the host controller has been selected and control transfer is used, the sequence bit should be set by software at a stage transition. For the ClearFeature request transmission or reception, the data PID sequence bit should be set by software regardless of whether the host controller or function controller is selected.

#### 25.3.4.8 Response PID = NAK Function

The USB has a function that disables pipe operation (PID response = NAK) at the timing at which the final data packet of a transaction is received (the USB automatically distinguishes this based on reception of a short packet or the transaction counter) by setting the PIPECFG.SHTNAK bit to 1.

When the double buffer mode is being used for the buffer memory, using this function enables reception of data packets in transfer units. If pipe operation has been disabled, software should set the pipe to the enabled state again (PID response = BUF).

The response PID = NAK function can be used only when bulk transfers are used.

#### 25.3.4.9 Auto Response Mode

With the pipes for bulk transfer (PIPE1 to PIPE5), when the PIPEnCTR.ATREPM bit is set to 1, a transition is made to auto response mode. During an OUT transfer (the PIPECFG.DIR bit = 0), OUT-NAK mode is entered, and during an IN transfer (the DIR bit = 1), null auto response mode is entered.

#### 25.3.4.10 OUT-NAK Mode

With the pipes for bulk OUT transfer, NAK is returned in response to an OUT token and an NRDY interrupt is output when the PIPEnCTR.ATREPM bit is set to 1. To make a transition from normal mode to OUT-NAK mode, OUT-NAK mode should be specified in the pipe operation disabled state (response PID = NAK) before enabling pipe operation (response PID = BUF). After pipe operation has been enabled, OUT-NAK mode becomes valid. However, if an OUT token is received immediately before pipe operation is disabled, the token data is normally received, and an ACK is retuned to the host.

To make a transition from OUT-NAK mode to normal mode, OUT-NAK mode should be canceled in the pipe operation disabled state (response PID = NAK) before enabling pipe operation (response PID = BUF). In normal mode, reception of OUT data is enabled.

#### 25.3.4.11 Null Auto Response Mode

With the pipes for bulk IN transfer, zero-length packets are continuously transmitted when the PIPEnCTR.ATREPM bit is set to 1.

To make a transition from normal mode to null auto response mode, null auto response mode should be set in the pipe operation disabled state (response PID = NAK) before enabling pipe operation (response PID = BUF). After pipe operation has been enabled, null auto response mode becomes valid. Before setting null auto response mode, the PIPEnCTR.INBUFM = 0 should be confirmed because the mode can be set only when the buffer is empty. If the INBUFM bit is 1, the buffer should be emptied with the PIPEnCTR.ACLRM bit. While a transition to null auto response mode is being made, data should not be written from the FIFO port.

To make a transition from null auto response mode to normal mode, pipe operation disabled state (response PID = NAK) should be retained for the period of zero-length packet transmission (about 10 µs) before canceling null auto response mode. In normal mode, data can be written from the FIFO port; therefore, packet transmission to the host is enabled by enabling pipe operation (response PID = BUF).

### 25.3.5 FIFO Buffer Memory

#### 25.3.5.1 FIFO Buffer Memory

The USB has FIFO buffer memory for data transfer. The memory area used for each pipe is managed by the USB. The FIFO buffer memory has two states depending on whether the access right is assigned to the system (CPU side) or the USB (SIE side).

##### (1) Buffer Status

Table 25.17 and Table 25.18 show the buffer status in the USB. The buffer memory status can be confirmed using the BSTS bit in DCPCTR and the INBUFM bit in PIPEnCTR. The transfer direction for the buffer memory can be specified using either the PIPECFG.DIR bit or the CFIFOSEL.ISEL bit (when DCP is selected).

The INBUFM bit is valid for PIPE0 to PIPE5 in the transmitting direction.

When a transmitting pipe uses the double buffer configuration, software can read the BSTS bit to monitor the buffer memory status on the CPU side and the INBUFM bit to monitor the buffer memory status on the SIE side. When the BEMP interrupt may not show the buffer empty status because the write access to the FIFO port by the CPU or DTC is slow, software can use the INBUFM bit to confirm the end of transmission.

**Table 25.17 Buffer Status Indicated by the BSTS Bit**

| ISEL or DIR | BSTS | Buffer Memory Status |
|---|---|---|
| 0 (receiving direction) | 0 | There is no received data, or data is being received.<br>Reading from the FIFO port is disabled. |
| 0 (receiving direction) | 1 | There is received data, or a zero-length packet has been received.<br>Reading from the FIFO port is allowed.<br>Note that when a zero-length packet is received, reading is not possible and the buffer must be cleared. |
| 1 (transmitting direction) | 0 | The transmission has not been completed.<br>Writing to the FIFO port is disabled. |
| 1 (transmitting direction) | 1 | The transmission has been completed.<br>CPU write is allowed. |

**Table 25.18 Buffer Status Indicated by the INBUFM Bit**

| DIR | INBUFM | Buffer Memory Status |
|---|---|---|
| 0 (receiving direction) | Invalid | Invalid |
| 1 (transmitting direction) | 0 | The transmission has been completed.<br>There is no waiting data to be transmitted. |
| 1 (transmitting direction) | 1 | The FIFO port has written data to the buffer.<br>There is data to be transmitted. |

### 25.3.5.2 FIFO Buffer Clearing

Table 25.19 shows the clearing of the FIFO buffer memory by the USB. The buffer memory can be cleared using the BCLR, DnFIFOSEL.DCLRM, and PIPEnCTR.ACLRM bit in the port control register.
Either a single or double buffer configuration can be selected for PIPE1 to PIPE5, using the PIPECFG.DBLB bit.

**Table 25.19 List of Buffer Clearing Methods**

| FIFO Buffer Clearing Mode | Clearing Buffer Memory on CPU Side | Mode for Automatically Clearing Buffer Memory after Reading Specified Pipe Data | Auto Buffer Clear Mode for Discarding All Received Packets |
|---|---|---|---|
| Register used | CFIFOCTR<br>DnFIFOCTR | DnFIFOSEL | PIPEnCTR |
| Bit used | BCLR | DCLRM | ACLRM |
| Clearing condition | Cleared by writing 1 | 1: Mode valid<br>0: Mode invalid | 1: Mode valid<br>0: Mode invalid |

#### (1) Auto Buffer Clear Mode Function

With the USB, all received data packets are discarded if the PIPEnCTR.ACLRM bit is set to 1. If a correct data packet has been received, the ACK response is returned to the host controller. The auto buffer clear mode function can be set only in the buffer memory reading direction.
If the ACLRM bit is set to 1 and then to 0, the buffer memory of the selected pipe can be cleared regardless of the access direction.
An access cycle of at least 100 ns is required for the internal hardware sequence processing time between ACLRM = 1 and ACLRM = 0.

### 25.3.5.3 FIFO Port Functions

Table 25.20 shows the settings for the FIFO port functions of the USB. In write access, writing data until the maximum packet size is reached automatically enables transmission of the data. To enable transmission before the maximum packet size is reached, the BVAL bit in the port control register should be set to end writing. To send a zero-length packet, the BCLR bit in the register should be used to clear the buffer and then the BVAL bit set in order to end writing.
In reading, reception of new packets is automatically enabled when all data has been read. Data cannot be read when a zero-length packet has been received (the DTLN[8:0] bits = 0), so the BCLR bit in the register should be used to clear the buffer. The length of the receive data can be confirmed using the DTLN[8:0] bits in the port control register.

**Table 25.20 FIFO Port Function Settings**

| Register Name | Bit Name | Description |
|---|---|---|
| CFIFOSEL, DnFIFOSEL (n = 0, 1) | RCNT | Selects DTLN read mode. |
| | REW | Buffer memory rewind (re-read, rewrite). |
| | DCLRM | Automatically clears receive data for a specified pipe after the data has been read (only for DnFIFO). |
| | DREQE | Enables DTC transfers (only for DnFIFO). |
| | MBW | FIFO port access bit width. |
| | BIGEND | Selects FIFO port endian. |
| | ISEL | FIFO port access direction (only for DCP). |
| | CURPIPE | Selects the current pipe. |
| CFIFOCTR, DnFIFOCTR (n = 0, 1) | BVAL | Ends writing to the buffer memory. |
| | BCLR | Clears the buffer memory on the CPU side. |
| | DTLN | Checks the length of receive data. |

#### (1) FIFO Port Selection

Table 25.21 shows the pipes that can be selected with the various FIFO ports. The pipe to be accessed should be selected using the CURPIPE[3:0] bits in the port select register. After the pipe is selected, whether the written value can be correctly read from the CURPIPE[3:0] bits should be checked. (If the previous pipe number is read, it indicates that the pipe modification is being executed by the USB controller.) Then, the FRDY bit in a port control register = 1 is checked.
In addition, the bus width to be accessed should be selected using the MBW bit in the port select register. The buffer memory access direction conforms to the PIPECFG.DIR bit. Only for the DCP, the ISEL bit in the port select register determines the direction.

**Table 25.21 FIFO Port Access Categorized by Pipe**

| Pipe | Access Method | Port that can be Used |
|---|---|---|
| DCP | CPU access | CFIFO port register |
| PIPE1 to PIPE9 | CPU access | CFIFO port register<br>D0FIFO/D1FIFO port register |
| | DTC access | D0FIFO/D1FIFO port register |

### (2) REW Bit

It is possible to temporarily stop access to the pipe currently being accessed, access a different pipe, and then continue processing for the current pipe again. The REW bit in the port select register is used for this processing.

If a pipe is selected through the CURPIPE[3:0] bits in the port select register with the REW bit set to 1, the pointer used for reading from and writing to the buffer memory is reset, and reading or writing can be carried out from the first byte.

If a pipe is selected with 0 set for the REW bit, data can be read and written in continuation from the previous selection, without the pointer being reset.

To access the FIFO port, the FRDY bit in the port control register = 1 should be checked after selecting a pipe.

### 25.3.6 Control Transfers Using DCP

In the data stage of control transfers, data is transferred using the default control pipe (DCP).
The DCP buffer memory is a 64-byte single buffer and is a fixed area that is shared for both control reading and control writing. The buffer memory can be accessed only through the CFIFO port.

#### 25.3.6.1 Control Transfers when Host Controller is Selected

##### (1) Setup Stage

USQREQ, USBVAL, USBINDX, and USBLENG are the registers that are used to transmit a USB request for setup transactions. Writing setup packet data to the registers and writing 1 to the DCPCTR.SUREQ bit transmits the specified data for setup transactions. Upon completion of the transaction, the SUREQ bit is cleared to 0. The above USB request registers should not be modified while SUREQ = 1.
After the attached state of the connected function device is detected, the first setup transaction for the device should be issued by using the sequence described above with the DCPMAXP.DEVSEL[3:0] bits set to 0 and the DEVADD0.USBSPD[1:0] bits set appropriately.
After the connected function device is shifted to the Address state, setup transactions should be issued by using the sequence described above with the assigned USB address set in the DEVSEL[3:0] bits and the bits in DEVADDn corresponding to the specified USB address set appropriately. For example, when PIPEMAXP.DEVSEL[3:0] = 0010b, make appropriate settings in DEVADD2; when PIPEMAXP.DEVSEL[3:0] = 0101b, make appropriate settings in DEVADD5.
When the setup transaction data has been sent, an interrupt request is generated according to the response received from the peripheral device (SIGN or SACK bit in INTSTS1), by means of which the result of the setup transactions can be confirmed.
A data packet of DATA0 (USB request) is transmitted as the data packet for a setup transaction regardless of the setting of the SQMON bit in DCPCTR.

##### (2) Data Stage

Data is transferred using the DCP buffer memory.
The access direction of the DCP buffer memory should be specified using the CFIFOSEL.ISEL bit. The transfer direction should be specified using the DCPCFG.DIR bit.
For the first data packet of the data stage, the data PID should be transferred as DATA1. Set data PID = DATA1 in the DCPCTR.SQSET bit and the PID bits = BUF in DCPCFG. Completion of data transfer is detected using the BRDY or BEMP interrupt.
For control write transfers, when the number of data bytes to be sent is an integer multiple of the maximum packet size, software should control so as to send a zero-length packet at the end.

##### (3) Status Stage

Zero-length packet data is transferred in the direction opposite to that in the data stage. As in the data stage, data is transferred using the DCP buffer memory. Transactions are done in the same manner as the data stage.
For the data packets of the status stage, the data PID should be set to DATA1 using the DCPCTR.SQSET bit.
For reception of a zero-length packet, the received data length should be confirmed using the CFIFOCTR.DTLN[8:0] after a BRDY interrupt is generated, and the buffer memory should then be cleared using the CFIFOCTR.BCLR bit.

### 25.3.6.2 Control Transfers when Function Controller is Selected

#### (1) Setup Stage

The USB always sends an ACK response for a correct setup packet targeted to the USB. The operation of the USB in the setup stage is described below.
When receiving a new setup packet, the USB sets the following bits.

- Set the INTSTS0.VALID bit to 1.
- Set the DCPCTR.PID[1:0] bits to NAK.
- Set the DCPCTR.CCPL bit to 0.

When receiving a data packet right after the setup packet, the USB stores the USB request parameters in USBREQ, USBVAL, USBINDX, and USBLENG.

Response processing with respect to the control transfer should always be carried out after setting the VALID bit = 0. In the VALID bit = 1 state, PID = BUF cannot be set, and the data stage cannot be terminated.
Using the function of the VALID bit, the USB can suspend the current request processing when receiving a new USB request during a control transfer, and can send a response to the newest request.
In addition, the USB automatically detects the direction bit (bit 8 of bmRequestType) and the request data length (wLength) of the received USB request, distinguishes between control read transfer, control write transfer, and no-data control transfer, and controls stage transitions. For a wrong sequence, the sequence error of the control transfer stage transition interrupt is generated, and the software is notified of occurrence of the error. For the stage control of the USB, see Figure 25.12.

#### (2) Data Stage

Data transfers corresponding to received USB requests should be done using the DCP. Before accessing the DCP buffer memory, the access direction should be specified using the CFIFOSEL.ISEL bit.
If the transfer data is larger than the size of the DCP buffer memory, the data transfer should be carried out using the BRDY interrupt for control write transfers and the BEMP interrupt for control read transfers.

#### (3) Status Stage

Control transfers are terminated by setting the DCPCTR.CCPL bit to 1 while the DCPCTR.PID[1:0] bits are set to BUF. After the above settings have been made, the USB automatically executes the status stage in accordance with the data transfer direction determined at the setup stage. The specific procedure is as follows.

- For control read transfers
  A zero-length packet is received from the USB host and an ACK response is sent.
- For control write transfers and no-data control transfers
  A zero-length packet is transmitted and an ACK response is received from the USB host.

#### (4) Control Transfer Auto Response Function

The USB automatically responds to a correct SET_ADDRESS request. If any of the following errors occurs in the SET_ADDRESS request, a response from the software is necessary.

- bmRequestType is not 00h: Any transfer other than a control write transfer
- wIndex is not 00h: Request error
- wIndex is not 00h: Any transfer other than a no-data control transfer
- wValue is larger than 7Fh: Request error
- INTSTS0.DVSQ[2:0] are 011b (configured): Control transfer of a device state error

For all requests other than the SET_ADDRESS request, a response is required from the corresponding software.

### 25.3.7 Bulk Transfers (PIPE1 to PIPE5)

The buffer memory usage (single/double buffer setting) can be selected for bulk transfers. The USB provides the following functions for bulk transfers.

- BRDY interrupt function (PIPECFG.BFRE bit: refer to section 25.3.3.1, (2) When the SOFCFG.BRDYM Bit = 0 and the PIPECFG.BFRE Bit = 1
- Transaction count function
  (PIPEnTRE.TRENB, TRCLR, and PIPEnTRN.TRNCNT[15:0] bits: refer to section 25.3.4.5, Transaction Counter (For PIPE1 to PIPE5 in Reading Direction)
- Response PID = NAK function (PIPECFG.SHTNAK bit: refer to section 25.3.4.8, Response PID = NAK Function
- Auto response mode (PIPEnCTR.ATREPM bit: refer to section 25.3.4.9, Auto Response Mode)

### 25.3.8 Interrupt Transfers (PIPE6 to PIPE9)

When the function controller is selected, the USB carries out interrupt transfers in accordance with the timing controlled by the host controller.
When the host controller is selected, the timing of issuing a token can be specified using the interval counter.

#### 25.3.8.1 Interval Counter during Interrupt Transfers when Host Controller is Selected

For interrupt transfers, intervals between transactions are set in the PIPEPERI.IITV[2:0] bits. The USB controller issues interrupt transfer tokens based on the specified intervals.

##### (1) Counter Initialization

The USB controller initializes the interval counter under the following conditions.

- Power-on reset:
  The IITV[2:0] bits are initialized.
- Buffer memory initialization using the PIPEnCTR.ACLRM bit:
  The IITV[2:0] bits are not initialized but the count value is initialized. Setting the PIPEnCTR.ACLRM bit to 0 starts counting from the value set in the IITV bits.

Note that the interval counter is not initialized in the following case.

- USB bus reset or USB suspended
  The IITV[2:0] bits are not initialized. Setting 1 to the UACT bit in DVSTCTR0 starts counting from the value before entering the USB bus reset state or USB suspended state.

##### (2) Operation when Transmission/Reception is Impossible at Token Issuance Timing

The USB cannot issue tokens even at token issuance timing in the following cases. In such a case, the USB attempts transactions at the subsequent interval.

- When the PID is set to NAK or STALL.
- When the buffer memory is full at the token sending timing in the receiving (IN) direction.
- When there is no data to be sent in the buffer memory at the token sending timing in the transmitting (OUT) direction.

### 25.3.9 Isochronous Transfers (PIPE1 and PIPE2)

The USB has the following functions for isochronous transfers.

- Notification of isochronous transfer error information
- Interval counter (specified by the PIPEPERI.IITV[2:0] bits)
- Isochronous IN transfer data setup control (IDLY function)
- Isochronous IN transfer buffer flush function (specified by the PIPEPERI.IFIS bit)

#### 25.3.9.1 Error Detection in Isochronous Transfers

The USB has a function for detecting the error information described below, so that when errors occur in isochronous transfers, they can be controlled by software. Table 25.22 and Table 25.23 show the priority in which errors are confirmed and the interrupts generated corresponding to errors.

##### (a) PID errors

- If the PID of the received packet is illegal.

##### (b) CRC errors and bit stuffing errors

- If an error occurs in the CRC of the received packet or the bit stuffing is illegal.

##### (c) Maximum packet size exceeded

- The data of the received packet is larger than the specified maximum packet size.

##### (d) Overrun and underrun errors

- When the host controller is selected
  When the buffer memory is full at the token sending timing in the IN (receiving) direction.
  When there is no data to be sent in the buffer memory at the token sending timing in the OUT (transmitting) direction.
- When the function controller is selected
  When there is no data to be sent in the buffer memory at the token receiving timing in the IN (transmitting) direction.
  When the buffer memory is full at the token receiving timing in the OUT (receiving) direction.

##### (e) Interval errors

An interval error is generated on any of the following conditions when the function controller is selected.

- During an isochronous IN transfer, an IN token could not be received in the interval frame.
- During an isochronous OUT transfer, an OUT token could not be received in the interval frame.

**Table 25.22 Error Detection When a Token is Received**

| Detection Priority | Error | Generated Interrupt and Status |
|---|---|---|
| 1 | PID errors | No interrupts are generated in both cases when the host controller is selected and the function controller is selected (ignored as a corrupted packet). |
| 2 | CRC errors and bit stuffing errors | No interrupts generated in both cases when the host controller is selected and the function controller is selected (ignored as a corrupted packet). |
| 3 | Overrun and underrun errors | An NRDY interrupt is generated to set the FRMNUM.OVRN bit to 1 in both cases when the host controller is selected and function controller is selected.<br>When the function controller is selected, a zero-length packet is transmitted in response to IN token. However, no data packets are received in response to OUT token. |
| 4 | Interval errors | An NRDY interrupt is generated when the function controller is selected. It is not generated when the host controller is selected. |

**Table 25.23 Error Detection When a Data Packet is Received**

| Detection Priority | Error | Generated Interrupt and Status |
|---|---|---|
| 1 | PID errors | No interrupts are generated (ignored as a corrupted packet). |
| 2 | CRC errors and bit stuffing errors | An NRDY interrupt is generated to set the FRMNUM.CRCE to 1 bit in both cases when the host controller is selected and the function controller is selected. |
| 3 | Maximum packet size exceeded errors | A BEMP interrupt is generated to set the PID[1:0] bits to STALL in both cases when the host controller is selected and the function controller is selected. |

### 25.3.9.2 DATA-PID

When the function controller is selected, the USB operates as follows in response to the received PID.
IN direction

- DATA0: Sent as data packet PID
- DATA1: Not sent
- DATA2: Not sent
- mDATA: Not sent

OUT direction

- DATA0: Received normally as data packet PID
- DATA1: Received normally as data packet PID
- DATA2: Packets are ignored
- mDATA: Packets are ignored

### 25.3.9.3 Interval Counter

The isochronous transfer interval can be set using the PIPEPERI.IITV[2:0] bits. The interval counter enables the functions shown in Table 25.24 when the function controller is selected. When the host controller is selected, the USB generates the token issuance timing. When the host controller is selected, the interval counter operation is the same as that in the interrupt transfer.

**Table 25.24 Interval Counter Function When the Function Controller is Selected**

| Transfer Direction | Function | Conditions for Detection |
|---|---|---|
| IN | Flushes transmit buffer | When an IN token cannot be successfully received in the interval frame during an isochronous IN transfer |
| OUT | Notifies that a token not being received | When an OUT token cannot be successfully received in the interval frame during an isochronous OUT transfer |

The interval count is carried out when an SOF is received or for interpolated SOFs, so the isochronism can be maintained even if an SOF is damaged. The frame interval that can be set is the $2^{IITV}$ frames.

### (1) Counter Initialization when Function Controller is Selected

The USB initializes the interval counter under the following conditions.

- Power-on Reset
  The PIPEPERI.IITV[2:0] bits are initialized.
- Buffer memory initialization using the ACLRM bit
  The IITV[2:0] bits are not initialized but the count value is initialized.

After the interval counter has been initialized, counting is started under either of the following conditions 1 and 2 when a packet has been transferred successfully.

1. An SOF is received after transmission of data in response to an IN token in the PID = BUF state.
2. An SOF is received after reception of data of an OUT token in the PID = BUF state.

Note that the interval counter is not initialized under the following conditions.

- When the PID[1:0] bits are set to NAK or STALL
  The interval timer does not stop. The USB attempts transactions at the subsequent interval.
- When the USB bus is reset or USB is suspended
  The IITV[2:0] bits are not initialized. When an SOF has been received, counting is restarted from the value prior to the reception of the SOF.

### (2) Interval Counting and Transfer Control when Host Controller is Selected

The USB controls the interval between token issuance operations based on the PIPEPERI.IITV[2:0] bit settings. Specifically, the USB issues a token for a selected pipe once every $2^{IITV}$ frames.
The USB starts counting the token issuance interval at the frame following the frame in which the PID[1:0] bits have been set to BUF by software.

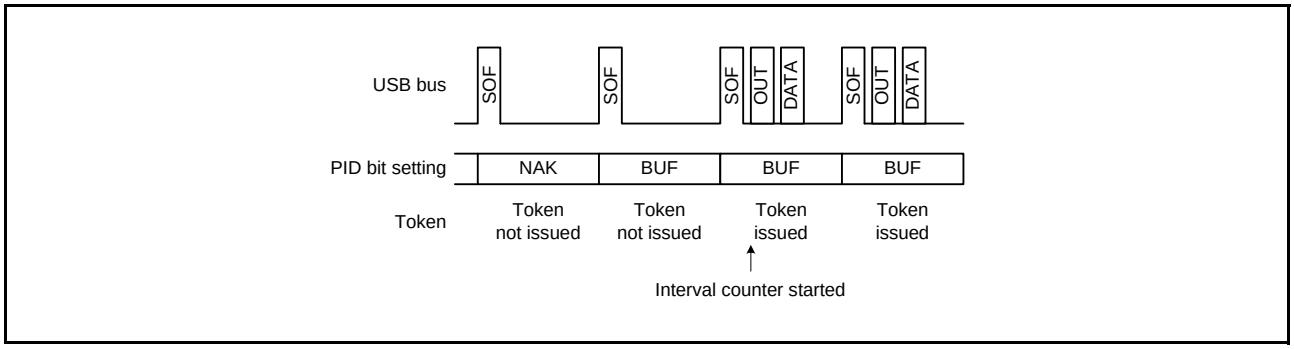



**Figure 25.13 Token Issuance When IITV = 0**

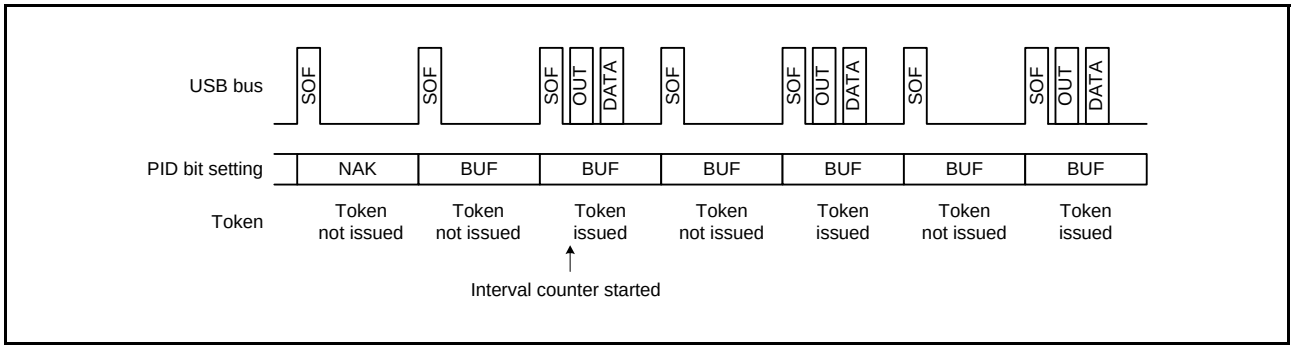



**Figure 25.14 Token Issuance When IITV = 1**

When the selected pipe is set for isochronous transfers, the USB carries out the following operation in addition to controlling the token issuance interval. The USB issues a token even when the NRDY interrupt generation condition is satisfied.

#### (a) When the selected pipe is for isochronous IN transfers

The USB generates an NRDY interrupt when the USB issues an IN token but does not receive a packet successfully from a peripheral device (no response or packet error).
The USB sets the FRMNUM.OVRN bit to 1 generating an NRDY interrupt when the time to issue an IN token comes while the USB cannot receive data because the FIFO buffer is full (due to the fact that the CPU or DTC is too slow to read data from the FIFO buffer).

#### (b) When the selected pipe is for isochronous OUT transfers

The USB sets the OVRN bit to 1 generating an NRDY interrupt and transmitting a zero-length packet when the time to issue an OUT token comes while there is no data to be transmitted in the FIFO buffer (because the CPU or DTC is too slow to write data to the FIFO buffer).
The token issuance interval is reset on any of the following conditions.

- When the USB is reset through a reset pin
  (The IITV[2:0] bits are also cleared to 0).
- When the PIPEnCTR.ACLRM bit has been set to 1 by software

### (3) Interval Counting and Transfer Control when Function Controller is Selected

#### (a) When the selected pipe is for isochronous OUT transfers

The USB generates an NRDY interrupt when the USB fails to receive a data packet within the interval set by the PIPEPERI.IITV[2:0] bits.
The USB also generates an NRDY interrupt when the USB fails to receive data because of a CRC error or other errors contained in the data packet or because of the FIFO buffer being full.
The NRDY interrupt is generated at the timing of SOF packet reception. Even if the SOF packet is corrupted, the internal interpolation allows the interrupt to be generated at the timing to receive the SOF packet.
However, when the IITV bit is set to a value other than 0, the USB generates an NRDY interrupt on receiving an SOF packet for every interval after starting interval counting operation.
When the PID[1:0] bits are set to NAK by software after starting the interval timer, the USB does not generate an NRDY interrupt on receiving an SOF packet.
The timing to start interval counting depends on the setting of IITV[2:0] bits as shown below.

- When the IITV = 0: The interval counting starts at the frame following the frame in which software has set the PID[1:0] bits for the selected pipe to BUF.

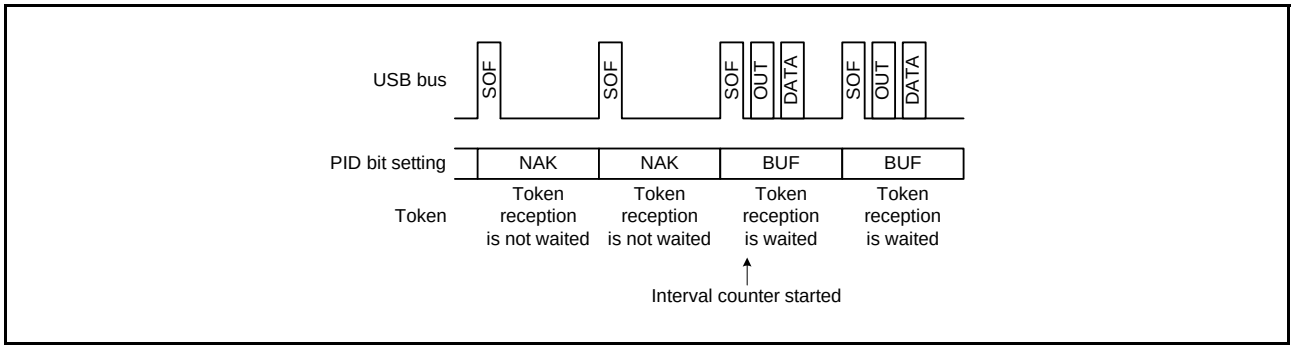



**Figure 25.15 Relationship between Frames and Expected Token Reception When IITV = 0**

- When the IITV ≠ 0: The interval counting starts on completion of successful reception of the first data packet after the PID[1:0] bits for the selected pipe have been modified to BUF.

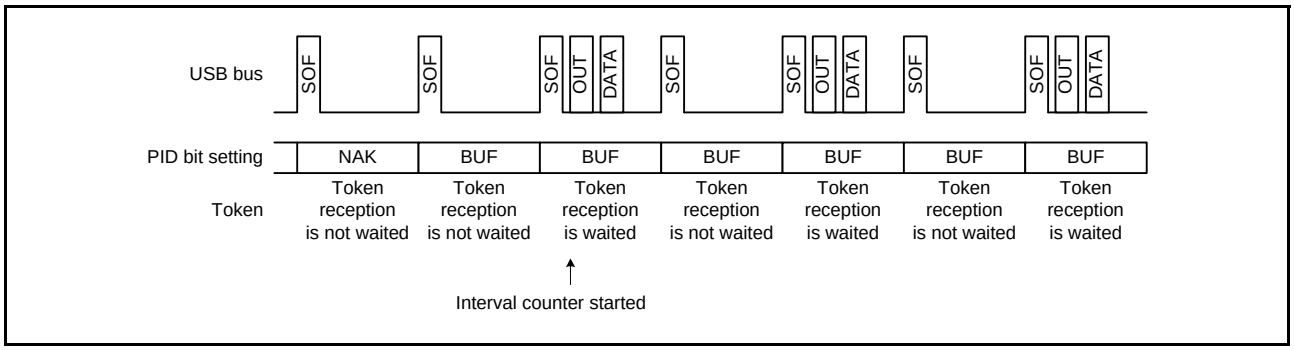



**Figure 25.16 Relationship between Frames and Expected Token Reception When IITV ≠ 0**

### (b) When the selected pipe is for isochronous IN transfers

The PIPEPERI.IFIS bit should be 1 for this use. When IFIS = 0, the USB transmits a data packet in response to the received IN token irrespective of the setting of the PIPEPERI.IITV[2:0] bits.
When IFIS = 1, the USB clears the FIFO buffer when the USB fails to receive an IN token in the frame at the interval set by the IITV[2:0] bits while there is data to be transmitted in the FIFO buffer.
The USB also clears the FIFO buffer when the USB fails to receive an IN token successfully because of a bus error such as a CRC error contained in the IN token.
The FIFO buffer is cleared at the timing of SOF packet reception. Even if the SOF packet is corrupted, the internal interpolation allows the FIFO buffer to be cleared at the timing to receive the SOF packet.

The timing to start interval counting depends on the setting of the IITV[2:0] bits (similar to the timing during OUT transfers).
The interval is counted on any of the following conditions in function controller mode.

- When a hardware-reset is applied to the USB (here, the IITV[2:0] bits are also cleared to 000b).
- When the PIPEnCTR.ACLRM bit is set to 1 by software.
- When the USB detects a USB bus reset.

### (4) Setup of Data to be Transmitted using Isochronous Transfer when Function Controller is Selected

With isochronous data transmission using the USB in the function controller, after data has been written to the buffer memory, a data packet can be transmitted with the next frame after the frame in which an SOF packet is detected. This function is called the isochronous transfer transmission data setup function, and it makes it possible to designate the frame from which transmission began.

In a double buffer configuration, even after the writing of data to both buffers has been completed, transmission will be enabled for only one buffer to which data writing was completed first. Accordingly, even if multiple IN tokens are received, only one packet of data is transmitted from a single buffer.

When an IN token is received, if the buffer memory is in the transmission enabled state, the USB transmits data as a normal response. If the buffer memory is not in the transmission enabled state, however, a zero-length packet is sent and an underrun error occurs.

Figure 25.17 shows an example of transmission using the isochronous transfer transmission data setup function with the USB when IITV = 0 (every frame) has been set.

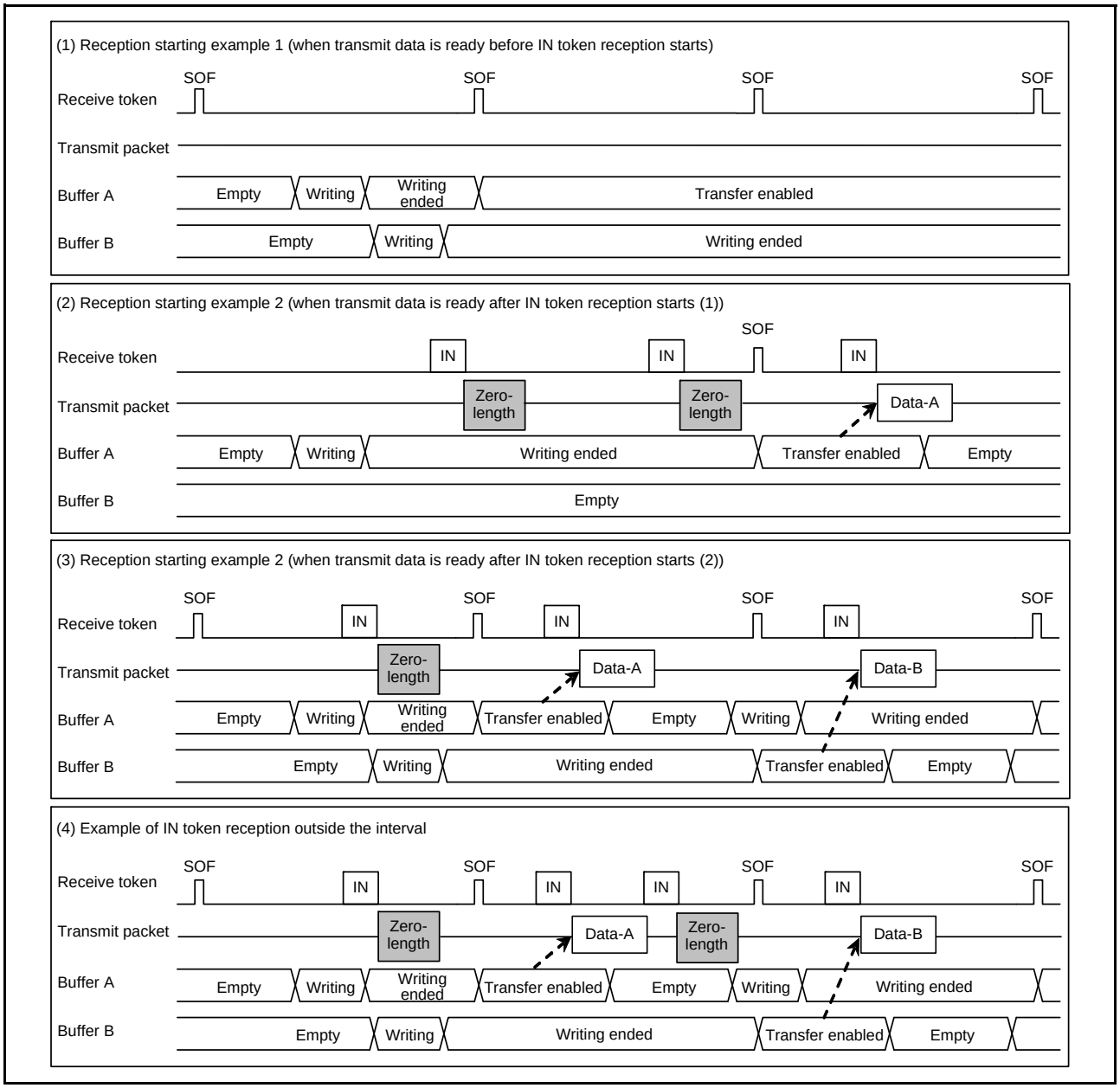



**Figure 25.17 Example of Data Setup Function Operation**

### (5) Isochronous Transfer Transmission Buffer Flush when Function Controller is Selected

If an SOF packet of the next frame is received without receiving an IN token in an interval frame during isochronous data transmission, the USB operates as if an IN token had been corrupted, and clears the buffer for which transmission is enabled, putting that buffer in the writing enabled state.
If a double buffer configuration is used and writing to both buffers has been completed, the buffer memory that was cleared is assumed as the data having been sent in the interval frame, and transmission is enabled for the buffer memory that is not cleared with SOF packet reception.
The timing of the buffer flush function depends on the setting of the PIPEPERI.IITV[2:0] bits.

- When the IITV = 0
  The buffer flush operation starts from the next frame after the pipe becomes valid.
- When the IITV ≠ 0
  The buffer flush operation is carried out after the first successful transaction.

Figure 25.18 shows an example of the buffer flush function in the USB. When an unanticipated token is received before the interval frame, the USB sends the write data or a zero-length packet as an underrun error according to the data setup state.

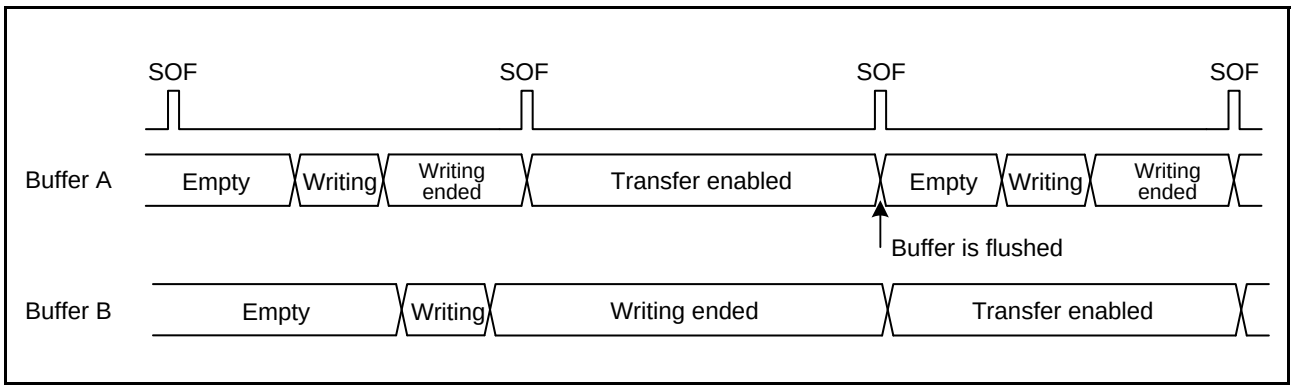



**Figure 25.18 Example of Buffer Flush Operation**

Figure 25.19 shows an example of interval error occurrence in the USB. There are five types of interval errors, as shown below. The interval error is generated at the timing indicated by (1) in the figure, and the buffer flush function is activated.

If an interval error occurs during an IN transfers, the buffer flush function is activated; if it occurs during an OUT transfer, an NRDY interrupt is generated.

The FRMNUM.OVRN bit should be used to distinguish between NRDY interrupts such as received packet errors and overrun errors.

In response to tokens that are shaded in the figure, responses are sent according to the buffer memory status.

IN direction

- If the buffer is in the transmission enabled state, the data is transferred as a normal response.
- If the buffer is in the transmission disabled state, a zero-length packet is sent and an underrun error occurs.

OUT direction

- If the buffer is in the reception enabled state, the data is received as a normal response.
- If the buffer is in the reception disabled state, the data is discarded and an overrun error occurs.

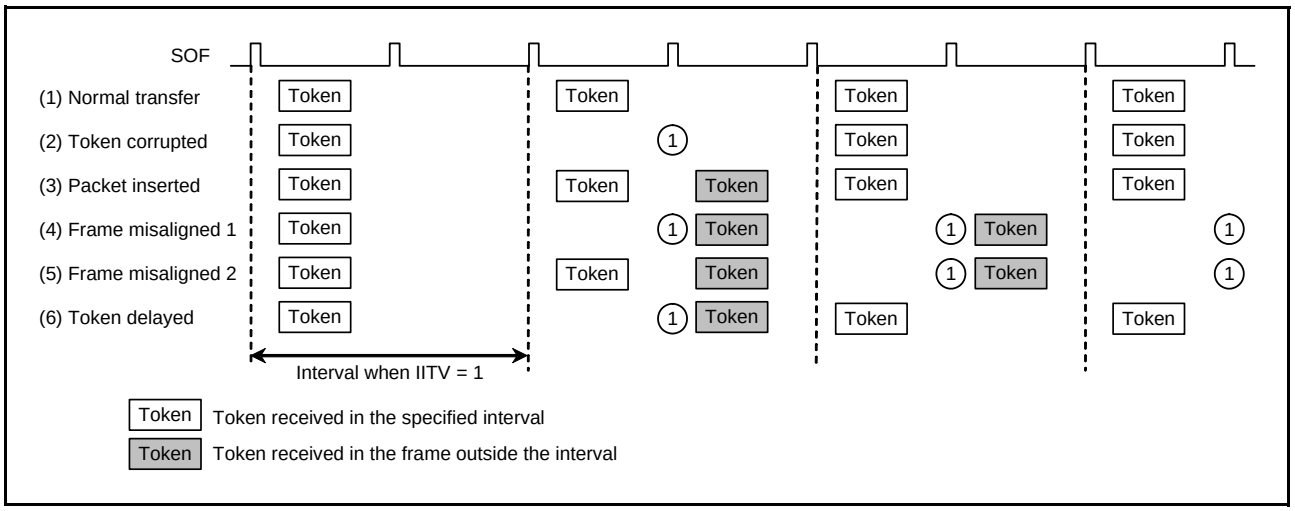



**Figure 25.19 Example of Interval Error Occurrence When IITV = 1**

### 25.3.10 SOF Interpolation Function

When the function controller is selected and if data could not be received at intervals of 1 ms because an SOF packet was corrupted or missing, the USB interpolates the SOF. The SOF interpolation operation begins when the USBE and SCKE bits in SYSCFG have been set to 1 and an SOF packet is received. The interpolation function is initialized under the following conditions.

- Power-on reset
- USB bus reset
- Suspended state detected

The SOF interpolation operates as follows.

- The interpolation function is not activated until an SOF packet is received.
- After the first SOF packet is received, interpolation is carried out by counting 1 ms with an internal clock of 48 MHz.
- After the second and subsequent SOF packets are received, interpolation is carried out at the previous reception interval.
- Interpolation is not carried out in the suspended state or while a USB bus reset is being received.

The USB supports the following functions based on the SOF packet reception. These functions also operate normally with SOF interpolation, if the SOF packet was missing.

- Updating of the frame number
- SOFR interrupt timing
- Isochronous transfer interval count

If an SOF packet is missing, the FRMNUM.FRNM[10:0] bits are not updated.

## 25.3.11 Pipe Schedule

### 25.3.11.1 Conditions for Generating a Transaction

When the host controller is selected and the DVSTCTR0.UACT bit has been set to 1, the USB generates a transaction under the conditions shown in Table 25.25.

**Table 25.25 Conditions for Generating a Transaction**

| | Conditions for Generation | | | | |
|---|---|---|---|---|---|
| **Transaction** | **DIR** | **PID** | **IITV0** | **Buffer State** | **SUREQ** |
| Setup | —*1 | —*1 | —*1 | —*1 | 1 setting |
| Control transfer data stage, status stage, bulk transfer | IN | BUF | Invalid | Receive area exists | —*1 |
| | OUT | BUF | Invalid | Transmit data exists | —*1 |
| Interrupt transfer | IN | BUF | Valid | Receive area exists | —*1 |
| | OUT | BUF | Valid | Transmit data exists | —*1 |
| Isochronous transfer | IN | BUF | Valid | *2 | —*1 |
| | OUT | BUF | Valid | *3 | —*1 |

Note 1. Symbols (—) in the table indicate that the condition is unrelated to the generating of tokens. "Valid" indicates that, for interrupt transfers and isochronous transfers, a transaction is generated only in transfer frames that are based on the interval counter. "Invalid" indicates that a transaction is generated regardless of the interval counter.
Note 2. This indicates that a transaction is generated regardless of whether there is a receive area. If there is no receive area, however, the received data is discarded.
Note 3. This indicates that a transaction is generated regardless of whether there is any data to be transmitted. If there is no data to be transmitted, however, a zero-length packet is transmitted.

### 25.3.11.2 Transfer Schedule

This section describes the transfer scheduling within a frame of the USB. After the USB sends an SOF, the transfer is carried out in the sequence described below.

1. Execution of periodic transfers
   A pipe is searched in the order of PIPE1 → PIPE2 → PIPE6 → PIPE7 → PIPE8 → PIPE9, and then, if there is a pipe for which an isochronous or interrupt transfer transaction can be generated, the transaction is generated.
2. Setup transactions for control transfers
   The DCP is checked, and if a setup transaction is possible, it is sent.
3. Execution of bulk transfers, control transfer data stages, and control transfer status stages
   A pipe is searched in the order of DCP → PIPE1 → PIPE2 → PIPE3 → PIPE4 → PIPE5, and then, if there is a pipe for which a transaction for a bulk transfer, a control transfer data stage, or a control transfer status stage can be generated, the transaction is generated.
   When a transaction is generated, processing moves to the next pipe transaction regardless of whether the response from the peripheral device is ACK or NAK. If there is time for transfer within the frame, step 3 is repeated.

### 25.3.11.3 Enabling USB Communication

Setting the DVSTCTR0.UACT bit to 1 initiates SOF transmission and transaction generation is enabled.
Setting the UACT bit to 0 stops SOF transmission and a suspend state is entered. If the setting of the UACT bit is changed from 1 to 0, processing stops after the next SOF is sent.

## 25.4 Usage Notes

### 25.4.1 Setting the Module-Stop Function

Operation of the USB module can be prohibited or permitted by a bit in module stop control register B (MSTPCRB). The setting after a reset is for operation of the USB module to be stopped. The registers are made accessible by release from the module-stop state. For details, refer to section 11., Low Power Consumption.

## 25.5 Battery Charging Detection Processing

It is possible to control the processing for data contact detection (D+ line contact check), primary detection (charger detection), and secondary detection (charger verification), which are defined in the battery charging specification.
The following describes required operations for a function device and a host device, individually.

### 25.5.1 Processing When Peripheral Controller is Selected

The following processing is required when operating the USB module as a portable device for battery charging.

(1) Detect when the data lines (D+/D–) have made contact and start the processing for primary detection.
(2) After primary detection starts, wait 40 ms for masking, and then check the D– voltage level to confirm the primary detection result.
(3) If the charger is detected during primary detection, also start secondary detection.
(4) After secondary detection starts, wait 40 ms for masking, and then check the D+ voltage level to confirm the secondary detection result.

For step (1), after VBUS is detected using the VBIT interrupt and the VBSTS bit, wait for 300 to 900 ms by software, and then set the VDPSRCE and IDMSINKE bits in the BCCTRL register. Or set the IDPSRCE bit, and after a change from high to low on the D+ line is detected using the LNST bits, clear the IDPSRCE bit and set the VDPSRCE and IDMSINKE bits. Set the VDPSRCE and IDMSINKE bits at the same time.*1

For step (2), set the VDPSRCE and IDMSINKE bits and wait 40 ms by software, and then use the CHGDETSTS bit to verify the primary detection result.*2

For step (3), if the CHGDETSTS bit is set in step (2), verify that the charger is detected, and then clear the VDPSRCE and IDMSINKE bits and set the VDMSRCE and IDPSINKE bits.

For step (4), set the VDMSRCE and IDPSINKE bits and wait for 40 ms by software, and then use the PDDETSTS bit to verify the secondary detection result.

The following shows the process flow.

Note 1. The battery charging specification describes two implementation methods of the process flow for data contact detection (D+/D– line contact check). One of the methods is to detect a change to logic low due to the pull-down resistor of the host device when the D+/D– lines have made contact with the target while the D+ line is held at logic high by applying a current of 7 to 13 μA on the D+ line. The other method is to wait for 300 to 900 ms after VBUS is detected.
Note 2. During primary detection, when the voltage on the D– line is detected to be 0.25 to 0.4 V or above and 0.8 to 2.0 V or below, the target device is recognized as the host device for battery charging (charging downstream port). When using a PHY in which the 0CHGDETSTS bit only indicates that the voltage on the D– line is 0.25 to 0.4 V or above, add the processing to check that the voltage on D– line is 0.8 V to 2.0 V or below using the LNST bits, as necessary.

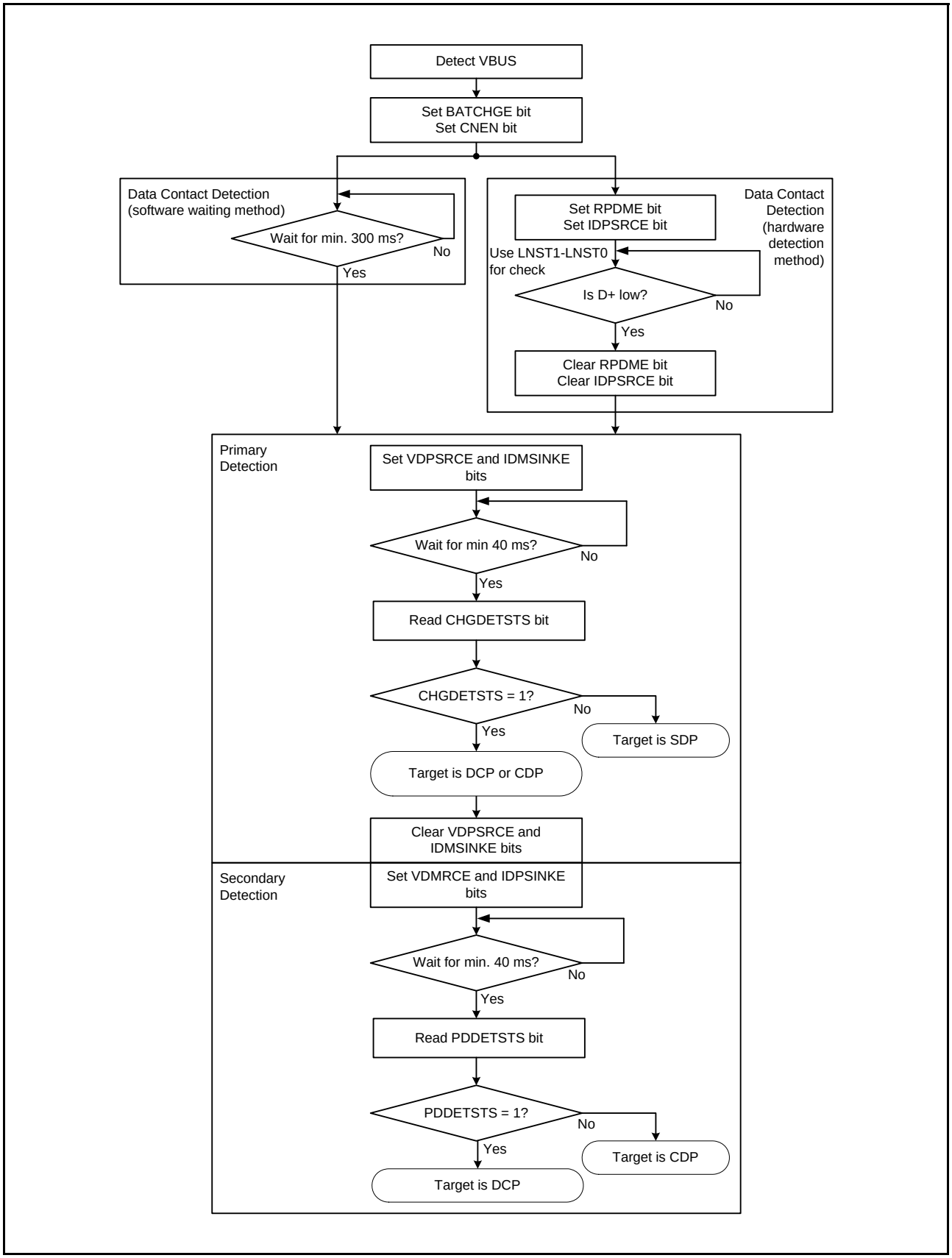



**Figure 25.20 Process Flow for Operating as Portable Device**

### 25.5.2 Processing When Host Controller is Selected

The following processing is required when operating the USB module as a charging downstream port for battery charging.

(1) Start driving the VBUS.
(2) Enable the portable device detection circuit.
(3) Monitor the portable device detection signal, and start driving the D– line if the detection signal is high.
(4) Detect when the portable device detection signal is low level and stop driving the D– line.

Or, the following processing can also be used in accordance with the battery charging specification.
(A) After disconnection is detected, start driving the D– line within 200 ms.
(B) After connection is detected, stop driving the D– line within 10 ms.

The D– line must be driven to allow the portable device to detect the primary detection described in section 25.5.1, Processing When Peripheral Controller is Selected. The above steps (1) to (4) apply when the portable device detection function is provided by hardware.
This method is to drive the D– line when the portable device is detected.
Steps (A) and (B) apply when the portable device function is not provided or available by hardware. Regardless of detection of the portable device, the D– line is driven in the disconnected state and the line is not driven in the connected state. In the battery charging specification, either of these methods can be used.

For steps (3) and (4), after a change in the portable device detection signal is detected using the PDDETINT interrupt, the current signal state can be confirmed by reading the PDDETSTS bit.

Steps (A) and (B) can be performed only in a software timer.

The following show the process flow for steps (1) to (4) and the process flow for steps (A) to (B), respectively.

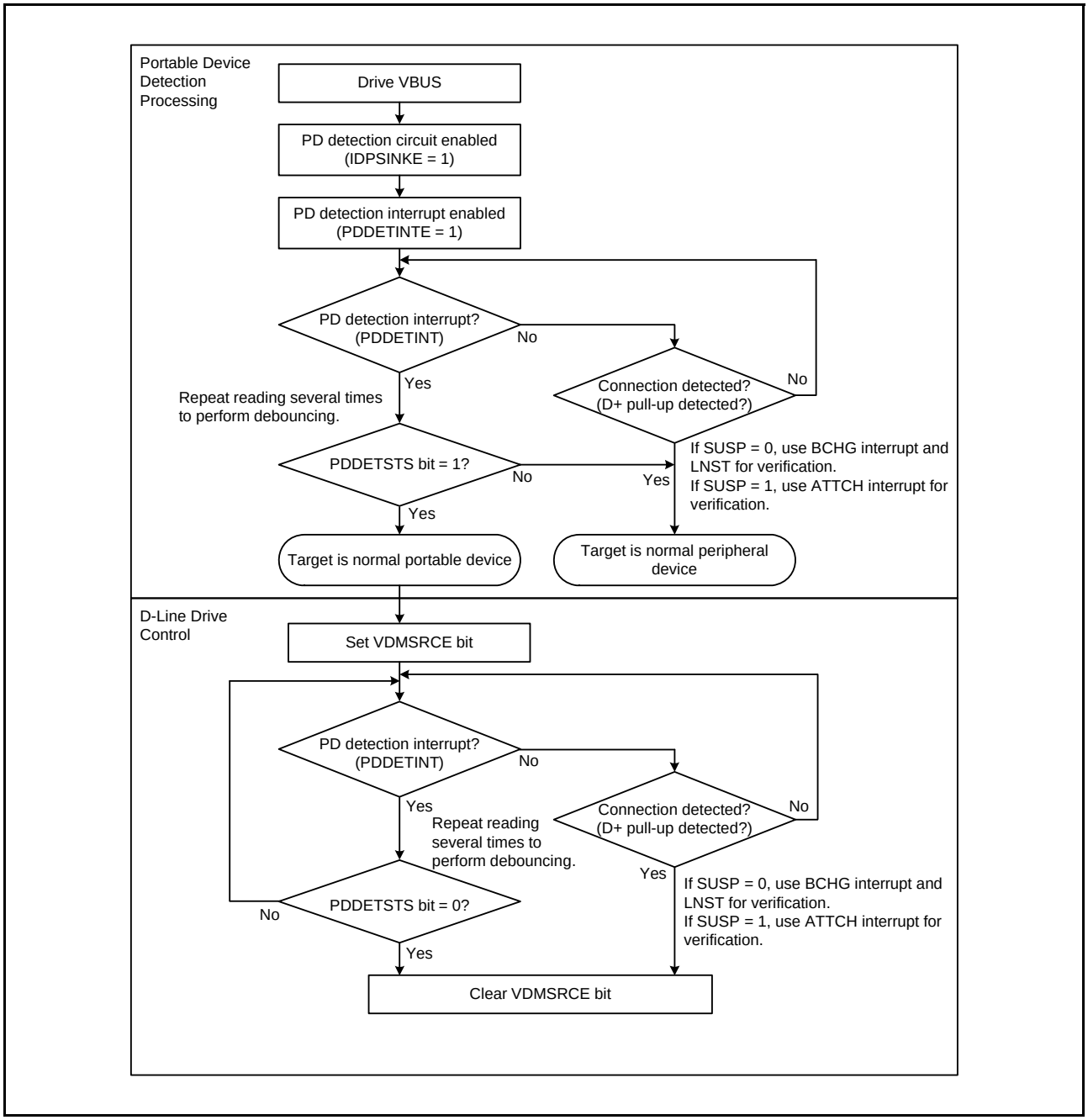



**Figure 25.21 Process Flow for Operating as Charging Downstream Port (Steps (1) to (4))**

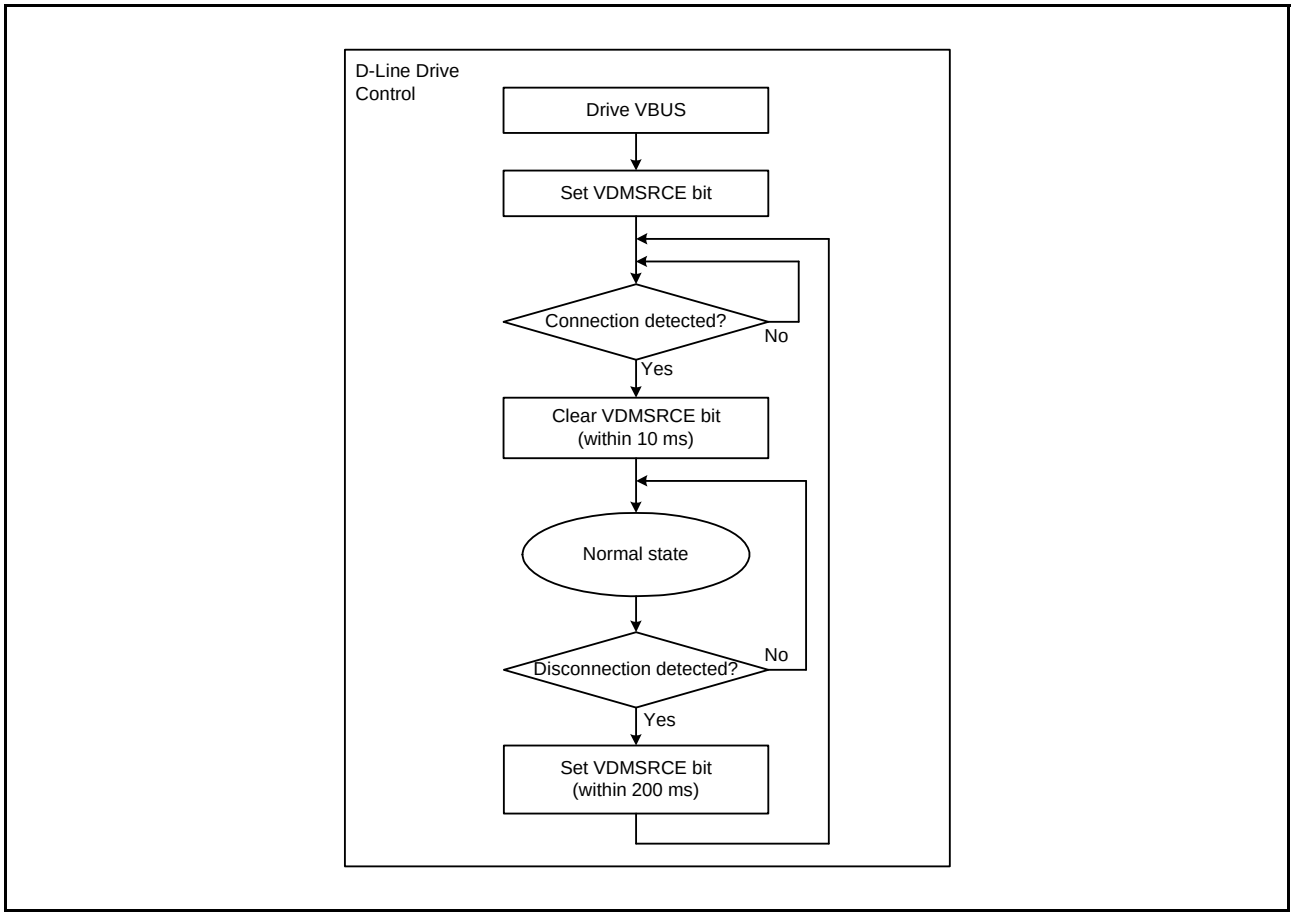



**Figure 25.22 Process Flow for Operating as Charging Downstream Port (Steps (A) to (B))**

# 26. Serial Communications Interface (SCIe, SCIf)

This MCU has three independent serial communications interface (SCI) channels. The SCI consists of the SCIe module (SCI1 and SCI5) and the SCIf module (SCI12).
The SCIe (SCI1 and SCI5) can handle both asynchronous and clock synchronous serial communications.
Asynchronous serial data communications can be carried out with standard asynchronous communications chips such as a Universal Asynchronous Receiver/Transmitter (UART) or Asynchronous Communications Interface Adapter (ACIA).
As an extended function in asynchronous communications mode, the SCI also supports smart card (IC card) interfaces conforming to ISO/IEC 7816-3 (standard for Identification Cards). The SCI is also supports simple SPI interfaces, and simple $I^2C$ bus interfaces when configured for single-master systems.
The SCIf module includes the functions of the SCIe module, and supports an extended serial communication protocol formed of Start Frames and Information Frames.

## 26.1 Overview

Table 26.1 lists the specifications of the SCIe module, Table 26.2 lists the specifications of the SCIf module, and Table 26.3 lists the specifications of the individual SCI channels.
Figure 26.1 and Figure 26.2 show block diagrams of the SCIe module, and Figure 26.3 shows the block diagram for the SCIf module.

**Table 26.1 Specifications of SCIe (1/2)**

| Item | | Description |
|---|---|---|
| Serial communication modes | | • Asynchronous operation<br>• Clock synchronous operation<br>• Smart card interface<br>• Simple $I^2C$ bus<br>• Simple SPI bus |
| Transfer speed | | Bit rate specifiable with the on-chip baud rate generator. |
| Full-duplex communications | | Transmitter: Enables continuous transmission by double-buffering.<br>Receiver: Enables continuous reception by double-buffering. |
| I/O pins | | See Table 26.4 to Table 26.6. |
| Data transfer | | Selectable as LSB first or MSB first transfer*1 |
| Interrupt sources | | Transmit end, transmit data empty, receive data full, and receive error<br>Completion of generation of a start condition, restart condition, or stop condition (for simple $I^2C$ mode) |
| Low power consumption function | | Module stop state can be set for each channel. |
| Asynchronous mode | Data length | 7 or 8 bits |
| | Transmission stop bit | 1 or 2 bits |
| | Parity | Even, odd, or none |
| | Receive error detection | Parity, overrun, and framing errors |
| | Hardware flow control | CTSn# and RTSn# pins can be used in transfer control. |
| | Start-bit detection | Low level or falling edge is selectable. |
| | Break detection | Break can be detected by reading RXDn pin level directly in case of a framing error |
| | Clock source | Selectable from internal or external clock<br>Enables transfer rate clock input from the MTU (SCI1 and SCI5) |
| | Multi-processor communications function | Serial communication among multiple processors |
| | Noise cancellation | The signal paths from input on the RXDn pins incorporate digital noise filters. |
| Clock synchronous mode | Data length | 8 bits |
| | Receive error detection | Overrun errors |
| | Hardware flow control | CTSn# and RTSn# pins can be used in transfer control. |

**Table 26.1 Specifications of SCIe (2/2)**

| Item | | Description |
|---|---|---|
| Smart card interface mode | Error processing | An error signal can be automatically transmitted when detecting a parity error during reception |
| | | Data can be automatically retransmitted when receiving an error signal during transmission |
| | Data type | Both direct convention and inverse convention are supported. |
| Simple $I^2C$ mode | Transfer format | $I^2C$ bus format |
| | Operating mode | Master (single-master operation only) |
| | Transfer rate | Up to 400 kbps |
| | Noise cancellation | The signal paths from input on the SSCLn and SSDAn pins incorporate digital noise filters, and the interval for noise cancellation is adjustable. |
| Simple SPI bus | Data length | 8 bits |
| | Detection of errors | Overrun errors |
| | SS input pin function | Applying the high level to the SSn# pin can cause the output pins to enter the high-impedance state. |
| | Clock settings | Four kinds of settings for clock phase and clock polarity are selectable. |
| Event link function (SCI5 only) | | Error (receive error or error signal detection) event output |
| | | Receive data full event output |
| | | Transmit data empty event output |
| | | Transmit end event output |

Note 1. In simple $I^2C$ mode, only MSB first is available.

**Table 26.2 Specifications of SCIf (1/2)**

| Item | | Description |
|---|---|---|
| Serial communication modes | | • Asynchronous operation<br>• Clock synchronous operation<br>• Smart card interface<br>• Simple $I^2C$ bus<br>• Simple SPI bus |
| Transfer speed | | Bit rate specifiable with the on-chip baud rate generator. |
| Full-duplex communications | | Transmitter: Enables continuous transmission by double-buffering.<br>Receiver: Enables continuous reception by double-buffering. |
| Input/output pins | | See Table 26.4 to Table 26.7. |
| Data transfer | | Selectable as LSB first or MSB first transfer*1 |
| Interrupt sources | | Transmit end, transmit data empty, receive data full, and receive error<br>Completion of generation of a start condition, restart condition, or stop condition (for simple $I^2C$ mode) |
| Low power consumption function | | Module stop state can be set. |
| Asynchronous mode | Data length | 7 or 8 bits |
| | Transmission stop bit | 1 or 2 bits |
| | Parity | Even, odd, or none |
| | Receive error detection | Parity, overrun, and framing errors |
| | Hardware flow control | CTSn# and RTSn# pins can be used in transfer control. |
| | Start-bit detection | Low level or falling edge is selectable. |
| | Break detection | Break can be detected by reading RXDn pin level directly in case of a framing error |
| | Clock source | Selectable from internal or external clock<br>Enables transfer rate clock input from the MTU (SCI12) |
| | Multi-processor communications function | Serial communication among multiple processors |
| | Noise cancellation | The signal paths from input on the RXDn pins incorporate digital noise filters. |

**Table 26.2 Specifications of SCIf (2/2)**

| Item | | Description |
|---|---|---|
| Clock synchronous mode | Data length | 8 bits |
| | Receive error detection | Overrun errors |
| | Hardware flow control | CTSn# and RTSn# pins can be used in transfer control. |
| Smart card interface mode | Error processing | An error signal can be automatically transmitted when detecting a parity error during reception |
| | | Data can be automatically retransmitted when receiving an error signal during transmission |
| | Data type | Both direct convention and inverse convention are supported. |
| Simple I2C mode | Transfer format | I2C bus format |
| | Operating mode | Master (single-master operation only) |
| | Transfer rate | Up to 400 kbps |
| | Noise cancellation | The signal paths from input on the SSCLn and SSDAn pins incorporate digital noise filters, and the interval for noise cancellation is adjustable. |
| Simple SPI bus | Data length | 8 bits |
| | Detection of errors | Overrun errors |
| | SS input pin function | Applying the high level to the SSn# pin can cause the output pins to enter the high-impedance state. |
| | Clock settings | Four kinds of settings for clock phase and clock polarity are selectable. |
| Extended serial mode | Start Frame transmission | • Output of a low level as the Break Field over a specified width and generation of interrupts on completion<br>• Detection of bus collisions and the generation of interrupts on detection |
| | Start Frame reception | • Detection of the Break Field low width and generation of an interrupt on detection<br>• Comparison of Control Fields 0 and 1 and generation of an interrupt when the two match<br>• Two kinds of data for comparison (primary and secondary) can be set in Control Field 1.<br>• A priority interrupt bit can be set in Control Field 1.<br>• Handling of Start Frames that do not include a Break Field<br>• Handling of Start Frames that do not include a Control Field<br>• Function for measuring bit rates |
| | I/O control function | • Selectable polarity for TXDX12 and RXDX12 signals<br>• Selection of a digital filter for RXDX12<br>• Half-duplex operation employing RXDX12 and TXDX12 signals multiplexed on the same pin<br>• Selectable timing for the sampling of data received through RXDX12<br>• Signals received on RXDX12 can be passed though to SCIe when the extended serial mode control section is off. |
| | Timer function | • Usable as a reloading timer |

Note 1. In simple I2C mode, only MSB first is available.

**Table 26.3 Functions of SCI Channels**

| Item | SCI1 | SCI5 | SCI12 |
|---|---|---|---|
| Asynchronous mode | Available | Available | Available |
| Clock synchronous mode | Available | Available | Available |
| Smart card interface mode | Available | Available | Available |
| Simple I2C mode | Available | Available | Available |
| Simple SPI bus | Available | Available | Available |
| Extended serial mode | Not available | Not available | Available |
| MTU clock input | Available | Available | Available |
| Event link function | Not available | Available | Not available |

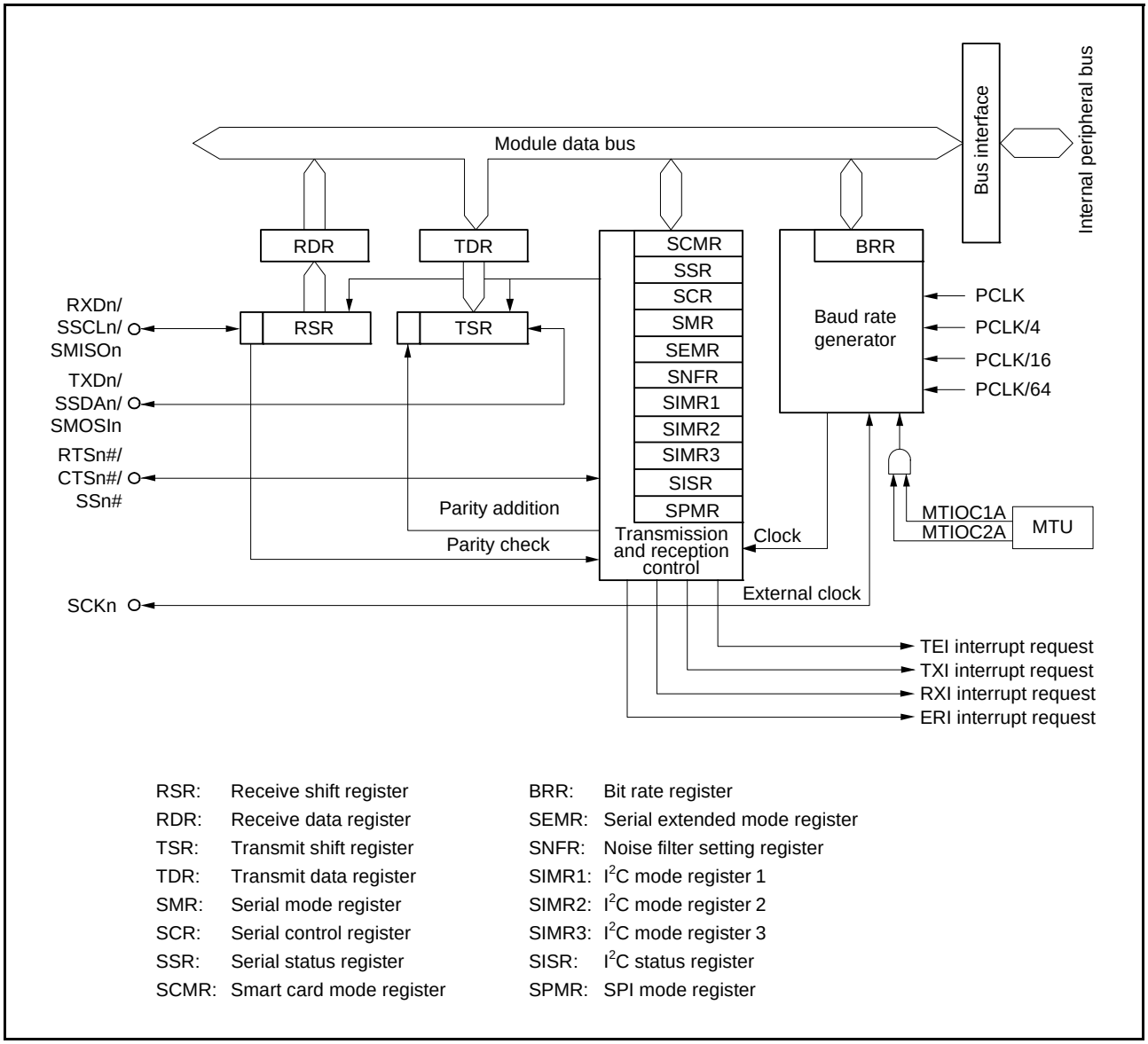



**Figure 26.1 Block Diagram of SCIe (SCI1)**

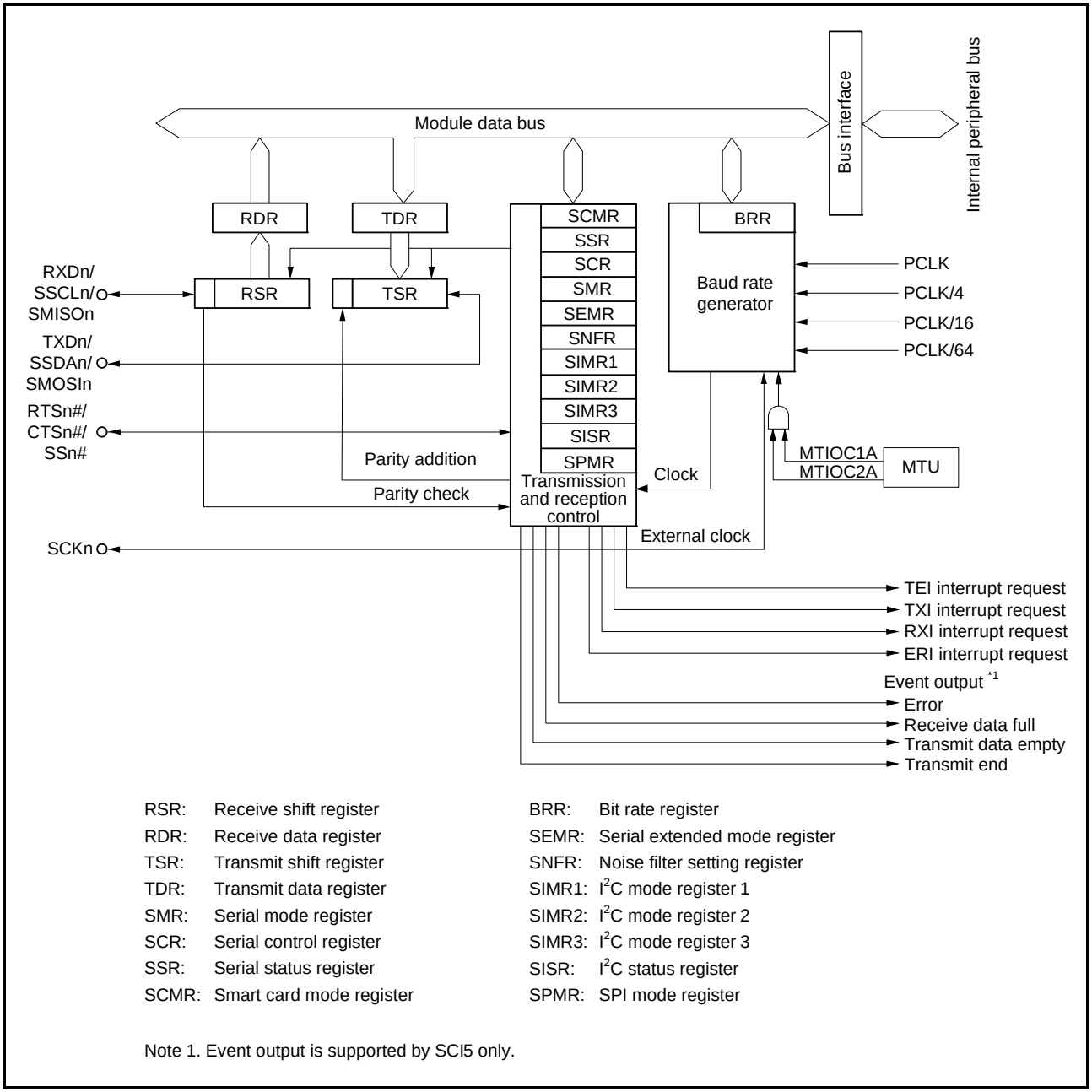



**Figure 26.2 Block Diagram of SCIe (SCI5)**

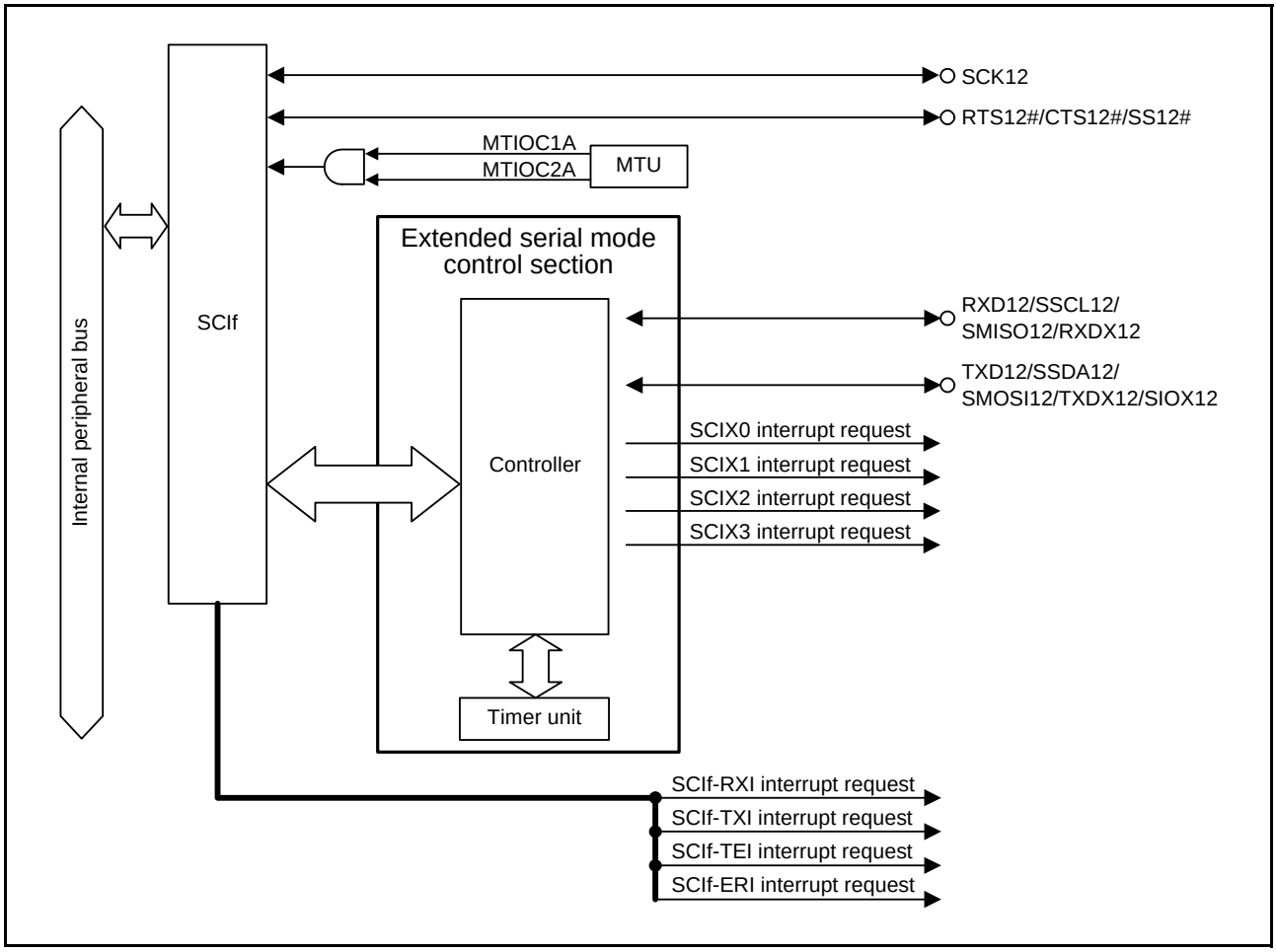



**Figure 26.3 Block Diagram of SCIf (SCI12)**

Table 26.4 to Table 26.7 list the pin configuration of the SCIs for the individual modes.

**Table 26.4 Pin Configuration of SCI (Asynchronous/Clock Synchronous Modes)**

| Channel | Pin Name | I/O | Function |
|---|---|---|---|
| SCI1 | SCK1 | I/O | SCI1 clock input/output |
| | RXD1 | Input | SCI1 receive data input |
| | TXD1 | Output | SCI1 transmit data output |
| | CTS1#/RTS1# | I/O | SCI1 transfer start control input/output |
| SCI5 | SCK5 | I/O | SCI5 clock input/output |
| | RXD5 | Input | SCI5 receive data input |
| | TXD5 | Output | SCI5 transmit data output |
| | CTS5#/RTS5# | I/O | SCI5 transfer start control input/output |
| SCI12 | SCK12 | I/O | SCI12 clock input/output |
| | RXD12 | Input | SCI12 receive data input |
| | TXD12 | Output | SCI12 transmit data output |
| | CTS12#/RTS12# | I/O | SCI12 transfer start control input/output |

**Table 26.5 Pin Configuration of SCI (Simple $I^2C$ Mode)**

| Channel | Pin Name | I/O | Function |
|---|---|---|---|
| SCI1 | SSCL1 | I/O | SCI1 $I^2C$ clock input/output |
| | SSDA1 | I/O | SCI1 $I^2C$ data input/output |
| SCI5 | SSCL5 | I/O | SCI5 $I^2C$ clock input/output |
| | SSDA5 | I/O | SCI5 $I^2C$ data input/output |
| SCI12 | SSCL12 | I/O | SCI12 $I^2C$ clock input/output |
| | SSDA12 | I/O | SCI12 $I^2C$ data input/output |

**Table 26.6 Pin Configuration of SCI (Simple SPI Mode)**

| Channel | Pin Name | I/O | Function |
|---|---|---|---|
| SCI1 | SCK1 | I/O | SCI1 clock input/output |
| | SMISO1 | I/O | SCI1 slave transmit data input/output |
| | SMOSI1 | I/O | SCI1 master transmit data input/output |
| | SS1# | Input | SCI1 chip select input |
| SCI5 | SCK5 | I/O | SCI5 clock input/output |
| | SMISO5 | I/O | SCI5 slave transmit data input/output |
| | SMOSI5 | I/O | SCI5 master transmit data input/output |
| | SS5# | Input | SCI5 chip select input |
| SCI12 | SCK12 | I/O | SCI12 clock input/output |
| | SMISO12 | I/O | SCI12 slave transmit data input/output |
| | SMOSI12 | I/O | SCI12 master transmit data input/output |
| | SS12# | Input | SCI12 chip select input |

**Table 26.7 Pin Configuration of SCI (Extended Serial Mode)**

| Channel | Pin Name | I/O | Function |
|---|---|---|---|
| SCI12 | RXDX12 | Input | SCI12 receive data input |
| | TXDX12 | Output | SCI12 transmit data output |
| | SIOX12 | I/O | SCI12 transfer data input/output |

## 26.2 Register Descriptions

### 26.2.1 Receive Shift Register (RSR)

RSR is a shift register which is used to receive serial data input from the RXDn pin and converts it into parallel data. When one frame of data has been received, it is transferred to RDR automatically.
RSR cannot be directly accessed by the CPU.

### 26.2.2 Receive Data Register (RDR)

Address(es): SCI1.RDR 0008 A025h, SCI5.RDR 0008 A0A5h, SCI12.RDR 0008 B305h

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

RDR is an 8-bit register that stores receive data.
When the SCI has received one frame of serial data, it transfers the received serial data from RSR register to RDR register. This allows RSR to receive the next data.
Since RSR and RDR function as a double buffer in this way, continuous receive operations can be performed.
Read RDR only once after a receive data full interrupt (RXI) has occurred. Note that if next one frame of data is received before reading receive data from RDR, an overrun error occurs.
RDR cannot be written to by the CPU.

### 26.2.3 Transmit Data Register (TDR)

Address(es): SCI1.TDR 0008 A023h, SCI5.TDR 0008 A0A3h, SCI12.TDR 0008 B303h

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| Value after reset: | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

TDR is an 8-bit register that stores transmit data.
When the SCI detects that TSR is empty, it transfers the transmit data written in TDR to TSR and starts transmission.
The double-buffered structures of TDR and TSR enable continuous serial transmission. If the next transmit data has already been written to TDR when one frame of data is transmitted, the SCI transfers the written data to TSR to continue transmission.
The CPU is able to read from or write to TDR at any time. Only write transmit data to TDR once after each instance of the transmit data empty interrupt (TXI).

### 26.2.4 Transmit Shift Register (TSR)

TSR is a shift register that transmits serial data.
To perform serial data transmission, the SCI first automatically transfers transmit data from TDR to TSR, and then sends the data to the TXDn pin.
TSR cannot be directly accessed by the CPU.

### 26.2.5 Serial Mode Register (SMR)

Note: • Some bits in SMR have different functions in smart card interface mode and non-smart card interface mode.

#### (1) Non-Smart Card Interface Mode (SCMR.SMIF = 0)

Address(es): SCI1.SMR 0008 A020h, SCI5.SMR 0008 A0A0h, SCI12.SMR 0008 B300h

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | CM | CHR | PE | PM | STOP | MP | CKS[1:0] | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b1, b0 | CKS[1:0] | Clock Select | b1 b0<br>0 0: PCLK clock (n = 0)*1<br>0 1: PCLK/4 clock (n = 1)*1<br>1 0: PCLK/16 clock (n = 2)*1<br>1 1: PCLK/64 clock (n = 3)*1 | R/W*4 |
| b2 | MP | Multi-Processor Mode | (Valid only in asynchronous mode)<br>0: Multi-processor communications function is disabled<br>1: Multi-processor communications function is enabled | R/W*4 |
| b3 | STOP | Stop Bit Length | (Valid only in asynchronous mode)<br>0: 1 stop bit<br>1: 2 stop bits | R/W*4 |
| b4 | PM | Parity Mode | (Valid only when the PE bit is 1)<br>0: Selects even parity<br>1: Selects odd parity | R/W*4 |
| b5 | PE | Parity Enable | (Valid only in asynchronous mode)<br>• When transmitting<br>0: Parity bit addition is not performed<br>1: The parity bit is added<br>• When receiving<br>0: Parity bit checking is not performed<br>1: The parity bit is checked | R/W*4 |
| b6 | CHR | Character Length | (Valid only in asynchronous mode)<br>0: Selects 8 bits as the data length*2<br>1: Selects 7 bits as the data length*3 | R/W*4 |
| b7 | CM | Communications Mode | 0: Asynchronous mode<br>1: Clock synchronous mode or simple SPI mode | R/W*4 |

Note 1. n is the decimal notation of the value of n in BRR (refer to section 26.2.9, Bit Rate Register (BRR)).
Note 2. In other than asynchronous mode, this bit setting is invalid and a fixed data length of 8 bits is used.
Note 3. LSB first is fixed and the MSB (bit 7) in TDR is not transmitted in transmission.
Note 4. Writable only when TE in SCR = 0 and RE in SCR = 0 (both serial transmission and reception are disabled).

**CKS[1:0] Bits (Clock Select)**

These bits select the clock source for the on-chip baud rate generator.
For the relation between the settings of these bits and the baud rate, refer to section 26.2.9, Bit Rate Register (BRR).

**MP Bit (Multi-Processor Mode)**

Disables/enables the multi-processor communications function. The settings of the PE bit and PM bit are invalid in multi-processor mode.

**STOP Bit (Stop Bit Length)**

Selects the stop bit length in transmission.
In reception, only the first stop bit is checked regardless of this bit setting. If the second stop bit is 0, it is treated as the start bit of the next transmit frame.

**PM Bit (Parity Mode)**

Selects the parity mode (even or odd) for transmission and reception.
The setting of the PM bit is invalid in multi-processor mode.

**PE Bit (Parity Enable)**

When this bit is set to 1, the parity bit is added to transmit data, and the parity bit is checked in reception.
Irrespective of the setting of the PE bit, the parity bit is not added or checked in multi-processor format.

**CHR Bit (Character Length)**

Selects the data length for transmission and reception.
In other than asynchronous mode, a fixed data length of 8 bits is used.

## (2) Smart Card Interface Mode (SCMR.SMIF = 1)

Address(es): SCI1.SMR 0008 A020h, SCI5.SMR 0008 A0A0h, SCI12.SMR 0008 B300h

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | GM | BLK | PE | PM | BCP[1:0] | | CKS[1:0] | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b1, b0 | CKS[1:0] | Clock Select | b1 b0<br>0 0: PCLK clock (n = 0)*1<br>0 1: PCLK/4 clock (n = 1)*1<br>1 0: PCLK/16 clock (n = 2)*1<br>1 1: PCLK/64 clock (n = 3)*1 | R/W*3 |
| b3, b2 | BCP[1:0] | Base Clock Pulse | Selects the number of base clock cycles in combination with the BCP2 bit in SCMR.<br>Setting values in BCP2 bit in SCMR and BCP[1:0] bits in SMR:<br>BCP2 BCP1 BCP0<br>0 0 0: 93 clock cycles (S = 93)*2<br>0 0 1: 128 clock cycles (S = 128)*2<br>0 1 0: 186 clock cycles (S = 186)*2<br>0 1 1: 512 clock cycles (S = 512)*2<br>1 0 0: 32 clock cycles (S = 32)*2 (Initial value)<br>1 0 1: 64 clock cycles (S = 64)*2<br>1 1 0: 372 clock cycles (S = 372)*2<br>1 1 1: 256 clock cycles (S = 256)*2 | R/W*3 |
| b4 | PM | Parity Mode | (Valid only when the PE bit is 1 in asynchronous mode)<br>0: Selects even parity<br>1: Selects odd parity | R/W*3 |
| b5 | PE | Parity Enable | When this bit is set to 1, a parity bit is added to transmit data, and the parity of received data is checked. Set this bit to 1 in smart card interface mode. | R/W*3 |
| b6 | BLK | Block Transfer Mode | 0: Normal mode operation<br>1: Block transfer mode operation | R/W*3 |
| b7 | GM | GSM Mode | 0: Normal mode operation<br>1: GSM mode operation | R/W*3 |

Note 1. n is the decimal notation of the value of n in BRR (refer to section 26.2.9, Bit Rate Register (BRR)).
Note 2. S is the value of S in BRR (refer to section 26.2.9, Bit Rate Register (BRR)).
Note 3. Writable only when TE in SCR = 0 and RE in SCR = 0 (both serial transmission and reception are disabled).

### CKS[1:0] Bits (Clock Select)

These bits select the clock source for the on-chip baud rate generator.

For the relationship between the settings of these bits and the baud rate, refer to section 26.2.9, Bit Rate Register (BRR).

### BCP[1:0] Bits (Base Clock Pulse)

These bits select the number of base clock cycles in a 1-bit data transfer time in smart card interface mode.

Set these bits in combination with the BCP2 bit in SCMR.

For details, refer to section 26.6.4, Receive Data Sampling Timing and Reception Margin.

### PM Bit (Parity Mode)

Selects the parity mode for transmission and reception (even or odd).

For details on the usage of this bit in smart card interface mode, refer to section 26.6.2, Data Format (Except in Block Transfer Mode).

**PE Bit (Parity Enable)**

Set the PE bit to 1.

The parity bit is added to transmit data before transmission, and the parity bit is checked in reception.

**BLK Bit (Block Transfer Mode)**

Setting this bit to 1 allows block transfer mode operation.

For details, refer to section 26.6.3, Block Transfer Mode.

**GM Bit (GSM Mode)**

Setting this bit to 1 allows GSM mode operation.

In GSM mode, the SSR.TEND flag set timing is put forward to 11.0 etu (elementary time unit = 1-bit transfer time) from the start and the clock output control function is appended. For details, refer to section 26.6.6, Serial Data Transmission (Except in Block Transfer Mode) and section 26.6.8, Clock Output Control.

## 26.2.6 Serial Control Register (SCR)

Note: • Some bits in SCR have different functions in smart card interface mode and non-smart card interface mode.

### (1) Non-Smart Card Interface Mode (SCMR.SMIF = 0)

Address(es): SCI1.SCR 0008 A022h, SCI5.SCR 0008 A0A2h, SCI12.SCR 0008 B302h

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | TIE | RIE | TE | RE | MPIE | TEIE | CKE[1:0] | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b1, b0 | CKE[1:0] | Clock Enable | (Asynchronous mode)<br>b1 b0<br>0 0: On-chip baud rate generator<br>The SCKn pin is available for use as an I/O port in accord with the I/O port settings.<br>0 1: On-chip baud rate generator<br>The clock with the same frequency as the bit rate is output from the SCKn pin.<br>1 x: External clock or MTU clock<br>• The clock with a frequency 16 times the bit rate should be input from the SCKn pin. Input a clock signal with a frequency eight times the bit rate when the SEMR.ABCS bit is 1.<br>• The MTU clock can be used.<br>The SCKn pin is available for use as an I/O port in accord with the I/O port settings when the MTU clock is used.<br><br>(Clock synchronous mode)<br>b1 b0<br>0 x: Internal clock<br>The SCKn pin functions as the clock output pin.<br>1 x: External clock<br>The SCKn pin functions as the clock input pin. | R/W*1 |
| b2 | TEIE | Transmit End Interrupt Enable | 0: A TEI interrupt request is disabled<br>1: A TEI interrupt request is enabled | R/W |
| b3 | MPIE | Multi-Processor Interrupt Enable | (Valid in asynchronous mode when SMR.MP = 1)<br>0: Normal reception<br>1: When the data with the multi-processor bit set to 0 is received, the data is not read, and setting the status flags ORER and FER in SSR to 1 is disabled. When the data with the multi-processor bit set to 1 is received, the MPIE bit is automatically cleared to 0, and normal reception is resumed. | R/W |
| b4 | RE | Receive Enable | 0: Serial reception is disabled<br>1: Serial reception is enabled | R/W*2 |
| b5 | TE | Transmit Enable | 0: Serial transmission is disabled<br>1: Serial transmission is enabled | R/W*2 |
| b6 | RIE | Receive Interrupt Enable | 0: RXI and ERI interrupt requests are disabled<br>1: RXI and ERI interrupt requests are enabled | R/W |
| b7 | TIE | Transmit Interrupt Enable | 0: A TXI interrupt request is disabled<br>1: A TXI interrupt request is enabled | R/W |

x: Don't care

Note 1. Writable only when TE = 0 and RE = 0.

Note 2. 1 can be written only when TE = 0 and RE = 0, while the SMR.CM bit is 1. After setting TE or RE to 1, only 0 can be written in TE and RE. While the SMR.CM bit is 0 and the SIMR1.IICM bit is 0, writing is enabled under any condition.

#### CKE[1:0] Bits (Clock Enable)

These bits select the clock source and SCKn pin function.
The combination of the settings of these bits and of the SEMR.ACS0 bit sets the internal MTU clock.

#### TEIE Bit (Transmit End Interrupt Enable)

Enables or disables a TEI interrupt request.
A TEI interrupt request is disabled by clearing the TEIE bit to 0.
In simple $I^2C$ mode, the TEI is allocated to the interrupt on completion of issuing a start, restart, or stop condition (STI). In this case, the TEIE bit can be used to enable or disable the STI.

#### MPIE Bit (Multi-Processor Interrupt Enable)

When this bit is set to 1 and the data with the multi-processor bit set to 0 is received, the data is not read and setting the status flags ORER and FER in SSR to 1 is disabled. When the data with the multi-processor bit set to 1 is received, the MPIE is automatically cleared to 0, and normal reception is resumed. For details, refer to section 26.4, Multi-Processor Communications Function.
When the receive data includes the MPB bit is SSR set to 0, the receive data is not transferred from the RSR to the RDR, a receive error is not detected, and setting the flags ORER and FER to 1 is disabled.
When the receive data includes the MPB bit set to 1, the MPB bit is set to 1, the MPIE bit is automatically cleared to 0, the RXI and ERI interrupt requests are enabled (if the RIE bit in SCR is set to 1), and setting the flags ORER and FER to 1 is enabled.
MPIE should be set to 0 if multi-processor communications function is not to be used.

#### RE Bit (Receive Enable)

Enables or disables serial reception.
When this bit is set to 1, serial reception is started by detecting the start bit in asynchronous mode or the synchronous clock input in clock synchronous mode. Note that SMR should be set prior to setting the RE bit to 1 in order to designate the reception format.
Even if reception is halted by clearing the RE bit to 0, the ORER, FER, and PER flags in SSR are not affected and the previous value is retained.

#### TE Bit (Transmit Enable)

Enables or disables serial transmission.
When this bit is set to 1, serial transmission is started by writing transmit data to TDR. Note that SMR should be set prior to setting the TE bit to 1 in order to designate the transmission format.

#### RIE Bit (Receive Interrupt Enable)

Enables or disables RXI and ERI interrupt requests.
An RXI interrupt request is disabled by clearing the RIE bit to 0.
An ERI interrupt request can be canceled by reading 1 from the ORER, FER, or PER flag in SSR and then clearing the flag to 0, or clearing the RIE bit to 0.

## (2) Smart Card Interface Mode (SCMR.SMIF = 1)

Address(es): SCI1.SCR 0008 A022h, SCI5.SCR 0008 A0A2h, SCI12.SCR 0008 B302h

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | TIE | RIE | TE | RE | MPIE | TEIE | CKE[1:0] | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b1, b0 | CKE[1:0] | Clock Enable | • When GM in SMR = 0<br>b1 b0<br>0 0: Output disabled<br>(The SCKn pin is available for use as an I/O port in accord with the I/O port settings.)<br>0 1: Clock output<br>1 x: (Setting prohibited)<br><br>• When GM in SMR = 1<br>b1 b0<br>0 0: Output fixed low<br>x 1: Clock output<br>1 0: Output fixed high | R/W*1 |
| b2 | TEIE | Transmit End Interrupt Enable | This bit should be 0 in smart card interface mode. | R/W |
| b3 | MPIE | Multi-Processor Interrupt Enable | This bit should be 0 in smart card interface mode. | R/W |
| b4 | RE | Receive Enable | 0: Serial reception is disabled<br>1: Serial reception is enabled | R/W*2 |
| b5 | TE | Transmit Enable | 0: Serial transmission is disabled<br>1: Serial transmission is enabled | R/W*2 |
| b6 | RIE | Receive Interrupt Enable | 0: RXI and ERI interrupt requests are disabled<br>1: RXI and ERI interrupt requests are enabled | R/W |
| b7 | TIE | Transmit Interrupt Enable | 0: A TXI interrupt request is disabled<br>1: A TXI interrupt request is enabled | R/W |

x: Don't care

Note 1. Writable only when TE = 0 and RE = 0.

Note 2. 1 can be written only when TE = 0 and RE = 0. After setting TE or RE to 1, only 0 can be written in TE and RE.

For details on interrupt requests, refer to section 26.11, Interrupt Sources.

### CKE[1:0] Bits (Clock Enable)

These bits control the clock output from the SCKn pin.

In GSM mode, clock output can be dynamically switched. For details, refer to section 26.6.8, Clock Output Control.

### TEIE Bit (Transmit End Interrupt Enable)

This bit should be 0 in smart card interface mode.

### MPIE Bit (Multi-Processor Interrupt Enable)

This bit should be 0 in smart card interface mode.

**RE Bit (Receive Enable)**

Enables or disables serial reception.

When this bit is set to 1, serial reception is started by detecting the start bit. Note that SMR should be set prior to setting the RE bit to 1 in order to designate the reception format.

Even if reception is halted by clearing the RE bit to 0, the ORER, FER, and PER flags in SSR are not affected and the previous value is retained.

**TE Bit (Transmit Enable)**

Enables or disables serial transmission.

When this bit is set to 1, serial transmission is started by writing transmit data to TDR. Note that SMR should be set prior to setting the TE bit to 1 in order to designate the transmission format.

**RIE Bit (Receive Interrupt Enable)**

Enables or disables RXI and ERI interrupt requests.

An RXI interrupt request is disabled by clearing the RIE bit to 0.

An ERI interrupt request can be canceled by reading 1 from the ORER, FER, or PER flag in SSR and then clearing the flag to 0, or clearing the RIE bit to 0.

## 26.2.7 Serial Status Register (SSR)

Note: • Some bits in SSR have different functions in smart card interface mode and non-smart card interface mode.

### (1) Non-Smart Card Interface Mode (SCMR.SMIF = 0)

Address(es): SCI1.SSR 0008 A024h, SCI5.SSR 0008 A0A4h, SCI12.SSR 0008 B304h

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | ORER | FER | PER | TEND | MPB | MPBT |
| Value after reset: | x | x | 0 | 0 | 0 | 1 | 0 | 0 |

x: Undefined

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b0 | MPBT | Multi-Processor Bit Transfer | Sets the multi-processor bit for adding to the transmission frame<br>0: Data transmission cycles<br>1: ID transmission cycles | R/W |
| b1 | MPB | Multi-Processor | Value of the multi-processor bit in the reception frame<br>0: Data transmission cycles<br>1: ID transmission cycles | R |
| b2 | TEND | Transmit End Flag | 0: A character is being transmitted.<br>1: Character transfer has been completed. | R |
| b3 | PER | Parity Error Flag | 0: No parity error occurred<br>1: A parity error has occurred | R/(W) *1 |
| b4 | FER | Framing Error Flag | 0: No framing error occurred<br>1: A framing error has occurred | R/(W) *1 |
| b5 | ORER | Overrun Error Flag | 0: No overrun error occurred<br>1: An overrun error has occurred | R/(W) *1 |
| b7, b6 | — | Reserved | The read value is undefined. The write value should be 1. | R/W |

Note 1. Only 0 can be written to this bit, to clear the flag.

#### MPB Bit (Multi-Processor)

Holds the value of the multi-processor bit in the reception frame. This bit does not change when the RE bit in SCR is 0.

#### TEND Flag (Transmission End Flag)

Indicates completion of transmission.

[Setting conditions]

- Clearing of the SCR.TE bit to 0 (serial transmission is disabled)
  When the SCR.TE bit is changed from 0 to 1, the TEND flag is not affected and retains the value 1.
- The TDR register is not updated at the time of transmission of the tail-end bit of a character being transmitted

[Clearing condition]

- When transmit data are written to the TDR register while the SCR.TE bit is 1
  When clearing the TEND flag to 0 to complete the interrupt handling, refer to section 14.4.1.2, Operation of Status Flags for Level-Detected Interrupts.

#### PER Flag (Parity Error Flag)

Indicates that a parity error has occurred during reception in asynchronous mode and the reception ends abnormally.

[Setting condition]

- When a parity error is detected during reception
  Although receive data when the parity error occurs is transferred to RDR, no RXI interrupt request occurs. Note that when the PER flag is being set to 1, the subsequent receive data is not transferred to RDR.

[Clearing condition]

- When 0 is written to PER after reading PER = 1 (after writing 0 to it, read the PER bit to check that it has actually been cleared to 0.)
  When clearing the PER flag to 0 to complete the interrupt handling, refer to section 14.4.1.2, Operation of Status Flags for Level-Detected Interrupts.
  Even when the RE bit in SCR is cleared to 0 (serial reception is disabled), the PER flag is not affected and retains its previous value.

#### FER Flag (Framing Error Flag)

Indicates that a framing error has occurred during reception in asynchronous mode and the reception ends abnormally.

[Setting condition]

- When the stop bit is 0
  In 2-stop-bit mode, only the first stop bit is checked whether it is 1 but the second stop bit is not checked. Note that although receive data when the framing error occurs is transferred to RDR, no RXI interrupt request occurs. In addition, when the FER flag is being set to 1, the subsequent receive data is not transferred to RDR.

[Clearing condition]

- When 0 is written to FER after reading FER = 1 (after writing 0 to it, read the FER bit to check that it has actually been cleared to 0.)
  When clearing the FER flag to 0 to complete the interrupt handling, refer to section 14.4.1.2, Operation of Status Flags for Level-Detected Interrupts.
  Even when the RE bit in SCR is cleared to 0, the FER flag is not affected and retains its previous value.

#### ORER Flag (Overrun Error Flag)

Indicates that an overrun error has occurred during reception and the reception ends abnormally.

[Setting condition]

- When the next data is received before receive data is read from RDR
  In RDR, receive data prior to an overrun error occurrence is retained, but data received after the overrun error occurrence is lost. When the ORER flag is set to 1, subsequent serial reception cannot be performed. Note that, in clock synchronous mode, serial transmission also cannot continue.

[Clearing condition]

- When 0 is written to ORER after reading ORER = 1 (after writing 0 to it, read the ORER bit to check that it has actually been cleared to 0.)
  When clearing the ORER flag to 0 to complete the interrupt handling, refer to section 14.4.1.2, Operation of Status Flags for Level-Detected Interrupts.
  Even when the RE bit in SCR is cleared to 0, the ORER flag is not affected and retains its previous value.

## (2) Smart Card Interface Mode (SCMR.SMIF = 1)

Address(es): SCI1.SSR 0008 A024h, SCI5.SSR 0008 A0A4h, SCI12.SSR 0008 B304h

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | ORER | ERS | PER | TEND | MPB | MPBT |
| Value after reset: | x | x | 0 | 0 | 0 | 1 | 0 | 0 |

x: Undefined

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b0 | MPBT | Multi-Processor Bit Transfer | This bit should be set to 0 in smart card interface mode. | R/W |
| b1 | MPB | Multi-Processor | This bit is not used in smart card interface mode. It should be set to 0. | R |
| b2 | TEND | Transmit End Flag | 0: A character is being transmitted.<br>1: Character transfer has been completed. | R |
| b3 | PER | Parity Error Flag | 0: No parity error occurred<br>1: A parity error has occurred | R/(W) *1 |
| b4 | ERS | Error Signal Status Flag | 0: Low error signal not responded<br>1: Low error signal responded | R/(W) *1 |
| b5 | ORER | Overrun Error Flag | 0: No overrun error occurred<br>1: An overrun error has occurred | R/(W) *1 |
| b7, b6 | — | Reserved | The read value is undefined. The write value should be 1. | R/W |

Note 1. Only 0 can be written to this bit, to clear the flag.

### MPBT Bit (Multi-Processor Bit Transfer)

This bit should be set to 0 in smart card interface mode.

### MPB Bit (Multi-Processor)

This bit is not used in smart card interface mode. It should be set to 0.

### TEND Flag (Transmission End Flag)

With no error signal from the receiving side, this bit is set to 1 when further data for transfer is ready to be transferred to the TDR register.

[Setting conditions]

- When the SCR.TE bit = 0 (serial transmission is disabled)
  When the SCR.TE bit is changed from 0 to 1, the TEND flag is not affected and retains the value 1.
- When a specified period has elapsed after the latest transmission of 1 byte, the ERS flag is 0, and the TDR register is not updated
  The set timing is determined by register settings as listed below.
  When SMR.GM = 0 and SMR.BLK = 0, 12.5 etu after the start of transmission
  When SMR.GM = 0 and SMR.BLK = 1, 11.5 etu after the start of transmission
  When SMR.GM = 1 and SMR.BLK = 0, 11.0 etu after the start of transmission
  When SMR.GM = 1 and SMR.BLK = 1, 11.0 etu after the start of transmission

[Clearing condition]

- When transmit data are written to the TDR register while the SCR.TE bit is 1

#### PER Flag (Parity Error Flag)

Indicates that a parity error has occurred during reception in asynchronous mode and the reception ends abnormally.

[Setting condition]

- When a parity error is detected during reception
  Although receive data when the parity error occurs is transferred to RDR, no RXI interrupt request occurs. Note that when the PER flag is being set to 1, the subsequent receive data is not transferred to RDR.

[Clearing condition]

- When 0 is written to PER after reading PER = 1 (After writing 0 to it, read the PER bit to check that it has actually been cleared to 0.)
  Even when the RE bit in SCR is cleared to 0 (serial reception is disabled), the PER flag is not affected and retains its previous value.

#### ERS Flag (Error Signal Status Flag)

[Setting condition]

- When a low error signal is sampled

[Clearing condition]

- When 0 is written to ERS after reading ERS = 1

#### ORER Flag (Overrun Error Flag)

Indicates that an overrun error has occurred during reception and the reception ends abnormally.

[Setting condition]

- When the next data is received before receive data is read from RDR
  In RDR, the receive data prior to an overrun error occurrence is retained, but data received following the overrun error occurrence is lost. When the ORER flag is set to 1, subsequent serial reception cannot be performed.

[Clearing condition]

- When 0 is written to ORER after reading ORER = 1 (After writing 0 to it, read the ORER bit to check that it has actually been cleared to 0.)
  Even when the RE bit in SCR is cleared to 0, the ORER flag is not affected and retains its previous value.

## 26.2.8 Smart Card Mode Register (SCMR)

Address(es): SCI1.SCMR 0008 A026h, SCI5.SCMR 0008 A0A6h, SCI12.SCMR 0008 B306h

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | BCP2 | — | — | — | SDIR | SINV | — | SMIF |
| Value after reset: | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b0 | SMIF | Smart Card Interface Mode Select | 0: Non-smart card interface mode (Asynchronous mode, clock synchronous mode, simple SPI mode, or simple I2C mode)<br>1: Smart card interface mode | R/W*1 |
| b1 | — | Reserved | This bit is read as 1. The write value should be 1. | R/W |
| b2 | SINV | Transmitted/Received Data Invert | 0: TDR contents are transmitted as they are. Receive data is stored as it is in RDR.<br>1: TDR contents are inverted before being transmitted. Receive data is stored in inverted form in RDR. | R/W*1 |
| b3 | SDIR | Transmitted/Received Data Transfer Direction | This bit can be used in the following modes.<br>• Asynchronous mode<br>• Clock synchronous mode<br>• Smart card interface mode<br>• Multiprocessor mode<br>• Simple SPI mode<br>Set this bit to 1 if operation is to be in simple I2C mode.<br>0: Transfer with LSB first<br>1: Transfer with MSB first | R/W*1 |
| b6 to b4 | — | Reserved | These bits are read as 1. The write value should be 1. | R/W |
| b7 | BCP2 | Base Clock Pulse 2 | Selects the number of base clock cycles in combination with the SMR.BCP[1:0] bits.<br>Setting values in the SCMR.BCP2 bit<br>BCP2 BCP1 BCP0<br>0 0 0: 93 clock cycles (S = 93)*2<br>0 0 1: 128 clock cycles (S = 128)*2<br>0 1 0: 186 clock cycles (S = 186)*2<br>0 1 1: 512 clock cycles (S = 512)*2<br>1 0 0: 32 clock cycles (S = 32)*2 (Initial Value)<br>1 0 1: 64 clock cycles (S = 64)*2<br>1 1 0: 372 clock cycles (S = 372)*2<br>1 1 1: 256 clock cycles (S = 256)*2 | R/W*1 |

Note 1. Writable only when TE in SCR = 0 and RE in SCR = 0 (both serial transmission and reception are disabled).
Note 2. S is the value of S in BRR (refer to section 26.2.9, Bit Rate Register (BRR)).

### SMIF Bit (Smart Card Interface Mode Select)

When this bit is set to 1, smart card interface mode is selected.

When this bit is set to 0, non-smart card interface mode, i.e., asynchronous mode (including multiprocessor mode), clock synchronous mode, simple SPI mode, or simple I2C mode is selected.

### SINV Bit (Transmitted/Received Data Invert)

Inverts the transmit/receive data logic level. This bit does not affect the logic level of the parity bit. To invert the parity bit, invert the PM bit in SMR.

### BCP2 Bit (Base Clock Pulse 2)

Selects the number of base clock cycles in a 1-bit data transfer time in smart card interface mode. Set this bit in combination with the SMR.BCP[1:0] bits.

## 26.2.9 Bit Rate Register (BRR)

Address(es): SCI1.BRR 0008 A021h, SCI5.BRR 0008 A0A1h, SCI12.BRR 0008 B301h

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| Value after reset: | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

BRR is an 8-bit register that adjusts the bit rate.
As each SCI channel has independent baud-rate generator control, different bit rates can be set for each. Table 26.8 shows the relationship between the setting (N) in the BRR and the bit rate (B) for normal asynchronous mode, multi-processor transfer, clock synchronous mode, smart card interface mode, simple SPI mode, and simple I2C mode.
The initial value of BRR is FFh.
BRR can be read from by the CPU at all times, but it can be written to only when the TE and RE bits in SCR are 0.

**Table 26.8 Relationship between N Setting in BRR and Bit Rate B**

| Mode | ABCS Bit in SEMR | BRR Setting | Error |
|---|---|---|---|
| Asynchronous, multi-processor transfer | 0 | $N = \frac{PCLK \times 10^6}{64 \times 2^{2n-1} \times B} - 1$ | $Error\ (\%) = \{\frac{PCLK \times 10^6}{B \times 64 \times 2^{2n-1} \times (N+1)} - 1\} \times 100$ |
| | 1 | $N = \frac{PCLK \times 10^6}{32 \times 2^{2n-1} \times B} - 1$ | $Error\ (\%) = \{\frac{PCLK \times 10^6}{B \times 32 \times 2^{2n-1} \times (N+1)} - 1\} \times 100$ |
| Clock synchronous, simple SPI | | $N = \frac{PCLK \times 10^6}{8 \times 2^{2n-1} \times B} - 1$ | |
| Smart card interface | | $N = \frac{PCLK \times 10^6}{S \times 2^{2n+1} \times B} - 1$ | $Error\ (\%) = \{\frac{PCLK \times 10^6}{B \times S \times 2^{2n+1} \times (N+1)} - 1\} \times 100$ |
| Simple $I^2C$ *1 | | $N = \frac{PCLK \times 10^6}{64 \times 2^{2n-1} \times B} - 1$ | |

B: Bit rate (bps)
N: BRR setting for baud rate generator ($0 \le N \le 255$)
PCLK: Operating frequency (MHz)
n and S: Determined by the settings of the SMR and SCMR registers as listed in the table below.
Note 1. Adjust the bit rate so that the widths at high and low level of the SCL output in simple I2C mode satisfy the I2C standard.

**Table 26.9 Calculating Widths at High and Low Level for SCL**

| Mode | SCL | Formula (Result in Seconds) |
|---|---|---|
| $I^2C$ | Width at high level (minimum value) | $(N+1) \times 4 \times 2^{2n-1} \times 7 \times \frac{1}{PCLK \times 10^6}$ |
| | Width at low level (minimum value) | $(N+1) \times 4 \times 2^{2n-1} \times 8 \times \frac{1}{PCLK \times 10^6}$ |

**Table 26.10 Clock Source Settings**

| SMR Setting | | |
|---|---|---|
| CKS[1:0] Bits | Clock Source | n |
| 0 0 | PCLK clock | 0 |
| 0 1 | PCLK/4 clock | 1 |

**Table 26.10 Clock Source Settings**

| SMR Setting | | |
|---|---|---|
| **CKS[1:0] Bits** | **Clock Source** | **n** |
| 1 0 | PCLK/16 clock | 2 |
| 1 1 | PCLK/64 clock | 3 |

**Table 26.11 Base Clock Settings in Smart Card Interface Mode**

| SCMR Setting | SMR Setting | | |
|---|---|---|---|
| **BCP2 Bit** | **BCP[1:0] Bits** | **Base Clock Cycles for 1-bit Period** | **S** |
| 0 | 0 0 | 93 clock cycles | 93 |
| 0 | 0 1 | 128 clock cycles | 128 |
| 0 | 1 0 | 186 clock cycles | 186 |
| 0 | 1 1 | 512 clock cycles | 512 |
| 1 | 0 0 | 32 clock cycles | 32 |
| 1 | 0 1 | 64 clock cycles | 64 |
| 1 | 1 0 | 372 clock cycles | 372 |
| 1 | 1 1 | 256 clock cycles | 256 |

Table 26.12 lists sample N settings in BRR in normal asynchronous mode. Table 26.13 lists the maximum bit rate settable for each operating frequency. Examples of BRR (N) settings in clock synchronous mode and simple SPI mode are listed in Table 26.16. Examples of BRR (N) settings in smart card interface mode are listed in Table 26.18. Examples of BRR (N) settings in simple $I^2C$ mode are listed in Table 26.21. In smart card interface mode, the number of base clock cycles S in a 1-bit data transfer time can be selected. For details, refer to section 26.6.4, Receive Data Sampling Timing and Reception Margin. Table 26.14 and Table 26.17 list the maximum bit rates with external clock input.

When the asynchronous mode base clock select bit (ABCS) in the serial extended mode register (SEMR) is set to 1 in asynchronous mode, the bit rate is two times that of listed in Table 26.12.

**Table 26.12 Examples of BRR Settings for Various Bit Rates (Asynchronous Mode)**

| Bit Rate (bps) | Operating Frequency PCLK (MHz) | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 8 | | | 9.8304 | | | 10 | | | 12 | | | 12.288 | | |
| | n | N | Error (%) | n | N | Error (%) | n | N | Error (%) | n | N | Error (%) | n | N | Error (%) |
| 110 | 2 | 141 | 0.03 | 2 | 174 | –0.26 | 2 | 177 | –0.25 | 2 | 212 | 0.03 | 2 | 217 | 0.08 |
| 150 | 2 | 103 | 0.16 | 2 | 127 | 0.00 | 2 | 129 | 0.16 | 2 | 155 | 0.16 | 2 | 159 | 0.00 |
| 300 | 1 | 207 | 0.16 | 1 | 255 | 0.00 | 2 | 64 | 0.16 | 2 | 77 | 0.16 | 2 | 79 | 0.00 |
| 600 | 1 | 103 | 0.16 | 1 | 127 | 0.00 | 1 | 129 | 0.16 | 1 | 155 | 0.16 | 1 | 159 | 0.00 |
| 1200 | 0 | 207 | 0.16 | 0 | 255 | 0.00 | 1 | 64 | 0.16 | 1 | 77 | 0.16 | 1 | 79 | 0.00 |
| 2400 | 0 | 103 | 0.16 | 0 | 127 | 0.00 | 0 | 129 | 0.16 | 0 | 155 | 0.16 | 0 | 159 | 0.00 |
| 4800 | 0 | 51 | 0.16 | 0 | 63 | 0.00 | 0 | 64 | 0.16 | 0 | 77 | 0.16 | 0 | 79 | 0.00 |
| 9600 | 0 | 25 | 0.16 | 0 | 31 | 0.00 | 0 | 32 | –1.36 | 0 | 38 | 0.16 | 0 | 39 | 0.00 |
| 19200 | 0 | 12 | 0.16 | 0 | 15 | 0.00 | 0 | 15 | 1.73 | 0 | 19 | –2.34 | 0 | 19 | 0.00 |
| 31250 | 0 | 7 | 0.00 | 0 | 9 | –1.70 | 0 | 9 | 0.00 | 0 | 11 | 0.00 | 0 | 11 | 2.40 |
| 38400 | — | — | — | 0 | 7 | 0.00 | 0 | 7 | 1.73 | 0 | 9 | –2.34 | 0 | 9 | 0.00 |

| Bit Rate (bps) | Operating Frequency PCLK (MHz) | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 14 | | | 16 | | | 17.2032 | | | 18 | | | 19.6608 | | |
| | n | N | Error (%) | n | N | Error (%) | n | N | Error (%) | n | N | Error (%) | n | N | Error (%) |
| 110 | 2 | 248 | –0.17 | 3 | 70 | 0.03 | 3 | 75 | 0.48 | 3 | 79 | –0.12 | 3 | 86 | 0.31 |
| 150 | 2 | 181 | 0.16 | 2 | 207 | 0.16 | 2 | 223 | 0.00 | 2 | 233 | 0.16 | 2 | 255 | 0.00 |
| 300 | 2 | 90 | 0.16 | 2 | 103 | 0.16 | 2 | 111 | 0.00 | 2 | 116 | 0.16 | 2 | 127 | 0.00 |
| 600 | 1 | 181 | 0.16 | 1 | 207 | 0.16 | 1 | 223 | 0.00 | 1 | 233 | 0.16 | 1 | 255 | 0.00 |
| 1200 | 1 | 90 | 0.16 | 1 | 103 | 0.16 | 1 | 111 | 0.00 | 1 | 116 | 0.16 | 1 | 127 | 0.00 |
| 2400 | 0 | 181 | 0.16 | 0 | 207 | 0.16 | 0 | 223 | 0.00 | 0 | 233 | 0.16 | 0 | 255 | 0.00 |
| 4800 | 0 | 90 | 0.16 | 0 | 103 | 0.16 | 0 | 111 | 0.00 | 0 | 116 | 0.16 | 0 | 127 | 0.00 |
| 9600 | 0 | 45 | –0.93 | 0 | 51 | 0.16 | 0 | 55 | 0.00 | 0 | 58 | –0.69 | 0 | 63 | 0.00 |
| 19200 | 0 | 22 | –0.93 | 0 | 25 | 0.16 | 0 | 27 | 0.00 | 0 | 28 | 1.02 | 0 | 31 | 0.00 |
| 31250 | 0 | 13 | 0.00 | 0 | 15 | 0.00 | 0 | 16 | 1.20 | 0 | 17 | 0.00 | 0 | 19 | –1.70 |
| 38400 | — | — | — | 0 | 12 | 0.16 | 0 | 13 | 0.00 | 0 | 14 | –2.34 | 0 | 15 | 0.00 |

| Bit Rate (bps) | Operating Frequency PCLK (MHz) | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | 20 | | | 25 | | | 30 | | |
| | n | N | Error (%) | n | N | Error (%) | n | N | Error (%) |
| 110 | 3 | 88 | –0.25 | 3 | 110 | –0.02 | 3 | 132 | 0.13 |
| 150 | 3 | 64 | 0.16 | 3 | 80 | 0.47 | 3 | 97 | –0.35 |
| 300 | 2 | 129 | 0.16 | 2 | 162 | –0.15 | 2 | 194 | 0.16 |
| 600 | 2 | 64 | 0.16 | 2 | 80 | 0.47 | 2 | 97 | –0.35 |
| 1200 | 1 | 129 | 0.16 | 1 | 162 | –0.15 | 1 | 194 | 0.16 |
| 2400 | 1 | 64 | 0.16 | 1 | 80 | 0.47 | 1 | 97 | –0.35 |
| 4800 | 0 | 129 | 0.16 | 0 | 162 | –0.15 | 0 | 194 | 0.16 |
| 9600 | 0 | 64 | 0.16 | 0 | 80 | 0.47 | 0 | 97 | –0.35 |
| 19200 | 0 | 32 | –1.36 | 0 | 40 | –0.76 | 0 | 48 | –0.35 |
| 31250 | 0 | 19 | 0.00 | 0 | 24 | 0.00 | 0 | 29 | 0 |
| 38400 | 0 | 15 | 1.73 | 0 | 19 | 1.73 | 0 | 23 | 1.73 |

Note: • This is an example when the ABCS bit in SEMR is 0.
When the ABCS bit is set to 1, the bit rate is two times.

**Table 26.13 Maximum Bit Rate for Each Operating Frequency (Asynchronous Mode)**

| PCLK (MHz) | Maximum Bit Rate (bps) | n | N |
|---|---|---|---|
| 8 | 250000 | 0 | 0 |
| 9.8304 | 307200 | 0 | 0 |
| 10 | 312500 | 0 | 0 |
| 12 | 375000 | 0 | 0 |
| 12.288 | 384000 | 0 | 0 |
| 14 | 437500 | 0 | 0 |
| 16 | 500000 | 0 | 0 |
| 17.2032 | 537600 | 0 | 0 |
| 18 | 562500 | 0 | 0 |
| 19.6608 | 614400 | 0 | 0 |
| 20 | 625000 | 0 | 0 |
| 25 | 781250 | 0 | 0 |
| 30 | 937500 | 0 | 0 |

Note: • When the ABCS bit in SEMR is set to 1, the bit rate is two times.

**Table 26.14 Maximum Bit Rate with External Clock Input (Asynchronous Mode)**

| | | Maximum Bit Rate (bps) | |
|---|---|---|---|
| PCLK (MHz) | External Input Clock (MHz) | SEMR.ABCS Bit = 0 | SEMR.ABCS Bit = 1 |
| 8 | 2.0000 | 125000 | 250000 |
| 9.8304 | 2.4576 | 153600 | 307200 |
| 10 | 2.5000 | 156250 | 312500 |
| 12 | 3.0000 | 187500 | 375000 |
| 12.288 | 3.0720 | 192000 | 384000 |
| 14 | 3.5000 | 218750 | 437500 |
| 16 | 4.0000 | 250000 | 500000 |
| 17.2032 | 4.3008 | 268800 | 537600 |
| 18 | 4.5000 | 281250 | 562500 |
| 19.6608 | 4.9152 | 307200 | 614400 |
| 20 | 5.0000 | 312500 | 625000 |
| 25 | 6.2500 | 390625 | 781250 |
| 30 | 7.5000 | 468750 | 937500 |

**Table 26.15 Maximum Bit Rate with MTU Clock Input (Asynchronous Mode)**

| | | Maximum Bit Rate (bps) | |
|---|---|---|---|
| PCLK (MHz) | MTU Clock (MHz) | SEMR.ABCS Bit = 0 | SEMR.ABCS Bit = 1 |
| 8 | 4 | 250000 | 500000 |
| 9.8304 | 4.9152 | 307200 | 614400 |
| 10 | 5 | 312500 | 625000 |
| 12 | 6 | 375000 | 750000 |
| 12.288 | 6.144 | 384000 | 768000 |
| 14 | 7 | 437500 | 875000 |
| 16 | 8 | 500000 | 1000000 |
| 17.2032 | 8.6016 | 537600 | 1075200 |
| 18 | 9 | 562500 | 1125000 |
| 19.6608 | 9.8304 | 614400 | 1228800 |
| 20 | 10 | 625000 | 1250000 |
| 25 | 12.5 | 781250 | 1562500 |
| 30 | 15 | 937500 | 1875000 |

**Table 26.16 BRR Settings for Various Bit Rates (Clock Synchronous Mode, Simple SPI Mode)**

| | Operating Frequency PCLK (MHz) | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 8 | | 10 | | 16 | | 20 | | 25 | | 30 | |
| Bit Rate (bps) | n | N | n | N | n | N | n | N | n | N | n | N |
| 110 | | | | | | | | | | | | |
| 250 | 3 | 124 | — | — | 3 | 249 | | | | | | |
| 500 | 2 | 249 | — | — | 3 | 124 | — | — | | | 3 | 233 |
| 1k | 2 | 124 | — | — | 2 | 249 | — | — | 3 | 97 | 3 | 116 |
| 2.5k | 1 | 199 | 1 | 249 | 2 | 99 | 2 | 124 | 2 | 155 | 2 | 187 |
| 5k | 1 | 99 | 1 | 124 | 1 | 199 | 1 | 249 | 2 | 77 | 2 | 93 |
| 10k | 0 | 199 | 0 | 249 | 1 | 99 | 1 | 124 | 1 | 155 | 1 | 187 |
| 25k | 0 | 79 | 0 | 99 | 0 | 159 | 0 | 199 | 0 | 249 | 1 | 74 |
| 50k | 0 | 39 | 0 | 49 | 0 | 79 | 0 | 99 | 0 | 124 | 0 | 149 |
| 100k | 0 | 19 | 0 | 24 | 0 | 39 | 0 | 49 | 0 | 62 | 0 | 74 |
| 250k | 0 | 7 | 0 | 9 | 0 | 15 | 0 | 19 | 0 | 24 | 0 | 29 |
| 500k | 0 | 3 | 0 | 4 | 0 | 7 | 0 | 9 | — | — | 0 | 14 |
| 1M | 0 | 1 | | | 0 | 3 | 0 | 4 | — | — | — | — |
| 2M | 0 | 0*1 | — | — | 0 | 1 | — | — | — | — | — | — |
| 2.5M | | | 0 | 0*1 | | | 0 | 1 | — | — | 0 | 2 |
| 4M | | | | | 0 | 0*1 | — | — | — | — | — | — |
| 5M | | | | | | | 0 | 0*1 | — | — | — | — |
| 6.25M | | | | | | | | | 0 | 0*1 | — | — |
| 7.5M | | | | | | | | | | | 0 | 0*1 |

Space: Setting prohibited.

—: Can be set, but an error will occur.

Note 1. Continuous transmission or reception is impossible. After transmitting/receiving one frame of data, there is an interval of a 1-bit period before starting transmitting/receiving the next frame of data. The output of the synchronization clock is stopped for a 1-bit period. For this reason, it takes 9 bits worth of time to transfer one frame (8 bits) of data, and the average transfer rate is $\frac{8}{9}$ times the bit rate.

**Table 26.17 Maximum Bit Rate with External Clock Input (Clock Synchronous Mode, Simple SPI Mode)**

| PCLK (MHz) | External Input Clock (MHz) | Maximum Bit Rate (Mbps) |
|---|---|---|
| 8 | 1.3333 | 1.3333 |
| 10 | 1.6667 | 1.6667 |
| 12 | 2.0000 | 2.0000 |
| 14 | 2.3333 | 2.3333 |
| 16 | 2.6667 | 2.6667 |
| 18 | 3.0000 | 3.0000 |
| 20 | 3.3333 | 3.3333 |
| 25 | 4.1667 | 4.1667 |
| 30 | 5.0000 | 5.0000 |

**Table 26.18 BRR Settings for Various Bit Rates (Smart Card Interface Mode, n = 0, S = 372)**

| Bit Rate (bps) | PCLK (MHz) | n | N | Error (%) |
|---|---|---|---|---|
| 9600 | 7.1424 | 0 | 0 | 0.00 |
| | 10.00 | 0 | 1 | 30 |
| | 10.7136 | 0 | 1 | 25 |
| | 13.00 | 0 | 1 | 8.99 |
| | 14.2848 | 0 | 1 | 0.00 |
| | 16.00 | 0 | 1 | 12.01 |
| | 18.00 | 0 | 2 | 15.99 |
| | 20.00 | 0 | 2 | 6.66 |
| | 25.00 | 0 | 3 | 12.49 |
| | 30.00 | 0 | 3 | 5.01 |

**Table 26.19 Maximum Bit Rate for Each Operating Frequency (Smart Card Interface Mode, S = 32)**

| PCLK (MHz) | Maximum Bit Rate (bps) | n | N |
|---|---|---|---|
| 10.00 | 156250 | 0 | 0 |
| 10.7136 | 167400 | 0 | 0 |
| 13.00 | 203125 | 0 | 0 |
| 16.00 | 250000 | 0 | 0 |
| 18.00 | 281250 | 0 | 0 |
| 20.00 | 312500 | 0 | 0 |
| 25.00 | 390625 | 0 | 0 |
| 30.00 | 468750 | 0 | 0 |

**Table 26.20 Maximum Bit Rate for Each Operating Frequency (Smart Card Interface Mode, S = 372)**

| PCLK (MHz) | Maximum Bit Rate (bps) | n | N |
|---|---|---|---|
| 10.00 | 13441 | 0 | 0 |
| 10.7136 | 14400 | 0 | 0 |
| 13.00 | 17473 | 0 | 0 |
| 16.00 | 21505 | 0 | 0 |
| 18.00 | 24194 | 0 | 0 |
| 20.00 | 26882 | 0 | 0 |
| 25.00 | 33602 | 0 | 0 |
| 30.00 | 40323 | 0 | 0 |

**Table 26.21 BRR Settings for Various Bit Rates (Simple I²C Mode)**

| Bit Rate (bps) | Operating Frequency PCLK (MHz) | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 8 | | | 10 | | | 16 | | | 20 | | | 25 | | |
| | n | N | Error (%) | n | N | Error (%) | n | N | Error (%) | n | N | Error (%) | n | N | Error (%) |
| 10 k | 0 | 24 | 0.0 | 0 | 31 | –2.3 | 1 | 12 | –3.8 | 1 | 15 | –2.3 | 1 | 19 | –2.3 |
| 25 k | 0 | 9 | 0.0 | 0 | 12 | –3.8 | 1 | 4 | 0.0 | 1 | 6 | –10.7 | 1 | 7 | –2.3 |
| 50 k | 0 | 4 | 0.0 | 0 | 6 | –10.7 | 1 | 2 | –16.7 | 1 | 3 | –21.9 | 1 | 3 | –2.3 |
| 100 k | 0 | 2 | –16.7 | 0 | 3 | –21.9 | 0 | 4 | 0.0 | 0 | 6 | –10.7 | 1 | 1 | –2.3 |
| 250 k | 0 | 0 | 0.0 | 0 | 1 | –37.5 | 0 | 1 | 0.0 | 0 | 2 | –16.7 | 0 | 3 | –21.9 |
| 350 k | | | | | | | | | | 0 | 1 | –10.7 | 0 | 2 | –25.6 |

| Bit Rate (bps) | Operating Frequency PCLK (MHz) | | |
|---|---|---|---|
| | 30 | | |
| | n | N | Error (%) |
| 10 k | 1 | 23 | –2.3 |
| 25 k | 1 | 9 | –6.3 |
| 50 k | 1 | 4 | –6.3 |
| 100 k | 1 | 2 | –21.9 |
| 250 k | 0 | 3 | –6.3 |
| 350 k | 0 | 2 | –10.7 |

**Table 26.22 Minimum Widths at High and Low Level for SCL at Various Bit Rates (Simple I²C Mode)**

| Bit Rate (bps) | Operating Frequency PCLK (MHz) | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 8 | | | 10 | | | 16 | | | 20 | | |
| | n | N | Min. Widths at High/Low Level for SCL (µs) | n | N | Min. Widths at High/Low Level for SCL (µs) | n | N | Min. Widths at High/Low Level for SCL (µs) | n | N | Min. Widths at High/Low Level for SCL (µs) |
| 10 k | 0 | 24 | 43.75/50.00 | 0 | 31 | 44.80/51.20 | 1 | 12 | 45.5/52.00 | 1 | 15 | 44.80/51.20 |
| 25 k | 0 | 9 | 17.50/20.00 | 0 | 12 | 18.2/20.80 | 1 | 4 | 17.50/20.00 | 1 | 6 | 19.60/22.40 |
| 50 k | 0 | 4 | 8.75/10.00 | 0 | 6 | 9.80/11.20 | 1 | 2 | 10.50/12.00 | 1 | 3 | 11.20/12.80 |
| 100 k | 0 | 2 | 5.25/6.00 | 0 | 3 | 5.60/6.40 | 0 | 4 | 4.37/5.00 | 0 | 6 | 4.90/5.60 |
| 250 k | 0 | 0 | 1.75/2.00 | 0 | 1 | 2.80/3.20 | 0 | 1 | 1.75/2.00 | 0 | 2 | 2.10/2.40 |
| 350 k | | | | | | | | | | 0 | 1 | 1.40/1.60 |

| Bit Rate (bps) | Operating Frequency PCLK (MHz) | | | | | |
|---|---|---|---|---|---|---|
| | 25 | | | 30 | | |
| | n | N | Min. Widths at High/Low Level for SCL (µs) | n | N | Min. Widths at High/Low Level for SCL (µs) |
| 10 k | 1 | 19 | 44.80/51.20 | 1 | 23 | 44.80/51.20 |
| 25 k | 1 | 7 | 17.92/20.48 | 1 | 9 | 18.66/21.33 |
| 50 k | 1 | 3 | 8.96/10.24 | 1 | 4 | 9.33/10.66 |
| 100 k | 1 | 1 | 4.48/5.12 | 1 | 2 | 5.60/6.40 |
| 250 k | 0 | 3 | 2.24/2.56 | 0 | 3 | 1.86/2.13 |
| 350 k | 0 | 2 | 1.68/1.92 | 0 | 2 | 1.40/1.60 |

## 26.2.10 Serial Extended Mode Register (SEMR)

Address(es): SCI1.SEMR 0008 A027h, SCI5.SEMR 0008 A0A7h, SCI12.SEMR 0008 B307h

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | RXDESEL | — | NFEN | ABCS | — | — | — | ACS0 |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b0 | ACS0 | Asynchronous Mode Clock Source Select | (Valid only in asynchronous mode)<br>0: External clock input<br>1: Logical AND of two clock cycles output from the MTU | R/W*1 |
| b3 to b1 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |
| b4 | ABCS | Asynchronous Mode Base Clock Select | (Valid only in asynchronous mode)<br>0: Selects 16 base clock cycles for 1-bit period<br>1: Selects 8 base clock cycles for 1-bit period | R/W*1 |
| b5 | NFEN | Digital Noise Filter Function Enable | (In asynchronous mode)<br>0: Noise cancellation function for the RXDn input signal is disabled.<br>1: Noise cancellation function for the RXDn input signal is enabled.<br>(in simple I2C mode)<br>0: Noise cancellation function for the SSCLn and SSDAn input signals is disabled.<br>1: Noise cancellation function for the SSCLn and SSDAn input signals is enabled.<br>The NFEN bit should be 0 in any mode other than above. | R/W*1 |
| b6 | — | Reserved | This bit is read as 0. The write value should be 0. | R/W |
| b7 | RXDESEL | Asynchronous Start Bit Edge Detection Select | (Valid only in asynchronous mode)<br>0: The low level on the RXDn pin is detected as the start bit.<br>1: A falling edge on the RXDn pin is detected as the start bit. | R/W*1 |

| SCI | MTU | Compare Match Output |
|---|---|---|
| SCI1 | MTU1, MTU2 | MTIOC1A, MTIOC2A |
| SCI5 | MTU1, MTU2 | MTIOC1A, MTIOC2A |
| SCI12 | MTU1, MTU2 | MTIOC1A, MTIOC2A |

Note 1. Writable only when TE in SCR = 0 and RE in SCR = 0 (both serial transmission and reception are disabled).

SEMR selects the clock source for 1-bit period in asynchronous mode.
The MTIOC1A and MTIOC2A output of the MTU can be set as the serial transfer base clock.
Figure 26.4 shows a setting example when the MTU clock input is selected.

### ACS0 Bit (Asynchronous Mode Clock Source Select)

Selects the clock source in the asynchronous mode.
The ACS0 bit is valid in asynchronous mode (CM bit in SMR = 0) and when an external clock input is selected (CKE[1:0] bits in SCR = 10b or 11b). An external clock input or internal MTU clock input can be selected.
Clear the ACS0 bit to 0 in other than asynchronous mode.

### NFEN Bit (Digital Noise Filter Function Enable)

This bit enables or disables the digital noise filter function.
When the function is enabled, noise cancellation is applied to the RXDn input signal in asynchronous mode, and noise cancellation is applied to the SSDAn and SSCLn input signals in simple I2C mode.
In any mode other than above, set the NFEN bit to 0 to disable the digital noise filter function.
When the function is disabled, input signals are transferred as is, as internal signals.

### RXDESEL Bit (Asynchronous Start Bit Edge Detection Select)

Selects the detection method of the start bit for reception in asynchronous mode. When a break occurs, data receiving operation depends on the settings of this bit. Set this bit to 1 when reception should be stopped while a break occurs or when reception should be started without retaining the RXDn pin input at high level for the period of one data frame or longer after completion of the break.

Set this bit to 0 in modes other than asynchronous mode.

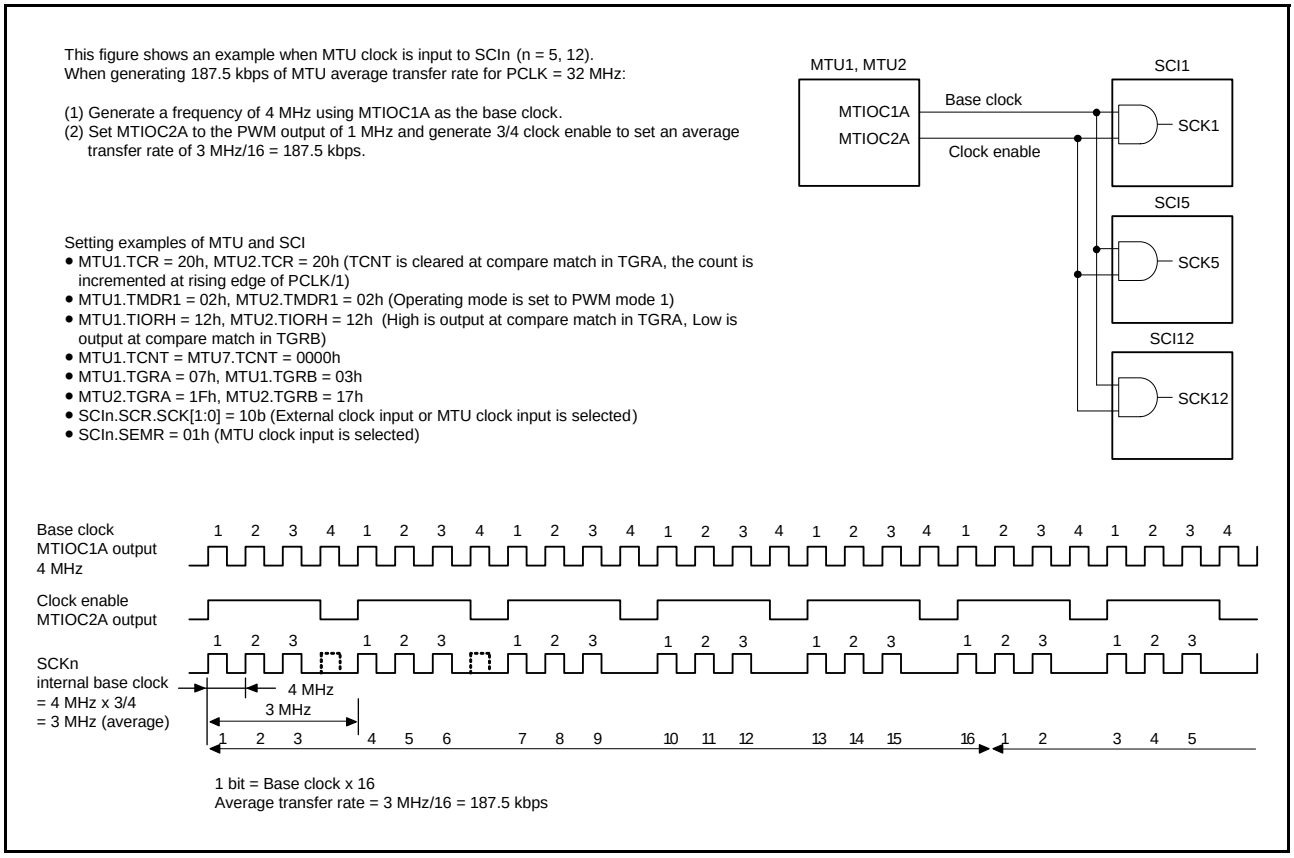



**Figure 26.4 Example of Average Transfer Rate Setting when MTU Clock is Input**

## 26.2.11 Noise Filter Setting Register (SNFR)

Address(es): SCI1.SNFR 0008 A028h, SCI5.SNFR 0008 A0A8h, SCI12.SNFR 0008 B308h

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | NFCS[2:0] | | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b2 to b0 | NFCS[2:0] | Noise Filter Clock Select | In asynchronous mode, the standard setting for the base clock is as follows.<br>b2 b0<br>0 0 0: The clock signal divided by 1 is used with the noise filter.<br><br>In simple I2C mode, the standard settings for the clock source of the on-chip baud rate generator selected by the SMR.CKS[1:0] bits are given below.<br>b2 b0<br>0 0 1: The clock signal divided by 1 is used with the noise filter.<br>0 1 0: The clock signal divided by 2 is used with the noise filter.<br>0 1 1: The clock signal divided by 4 is used with the noise filter.<br>1 0 0: The clock signal divided by 8 is used with the noise filter.<br><br>Settings other than above are prohibited. | R/W*1 |
| b7 to b3 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |

Note 1. Writing to these bits is only possible when the RE and TE bits in the SCR are 0 (serial reception and transmission disabled).

### NFCS[2:0] Bits (Noise Filter Clock Select)

These bits select the sampling clock for the digital noise filter. To use the noise filter in asynchronous mode, set these bits to 000b. In simple I2C mode, set the bits to a value in the range from 001b to 100b.

## 26.2.12 I²C Mode Register 1 (SIMR1)

Address(es): SCI1.SIMR1 0008 A029h, SCI5.SIMR1 0008 A0A9h, SCI12.SIMR1 0008 B309h

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | IICDL[4:0] | | | | | — | — | IICM |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b0 | IICM | Simple I²C Mode Select | SMIF IICM<br>0 0: Asynchronous mode, Multiprocessor mode, Clock synchronous mode (in asynchronous mode, synchronous, or simple SPI mode)<br>0 1: Simple I²C mode<br>1 0: Smart card interface mode<br>1 1: Setting prohibited. | R/W*1 |
| b2, b1 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |
| b7 to b3 | IICDL[4:0] | SSDA Delay Output Select | (Cycles below are of the clock signal from the on-chip baud rate generator.)<br>b7 b3<br>0 0 0 0 0: No output delay<br>0 0 0 0 1: 0 to 1 cycle<br>0 0 0 1 0: 1 to 2 cycles<br>0 0 0 1 1: 2 to 3 cycles<br>0 0 1 0 0: 3 to 4 cycles<br>0 0 1 0 1: 4 to 5 cycles<br>: :<br>1 1 1 1 0: 29 to 30 cycles<br>1 1 1 1 1: 30 to 31 cycles | R/W*1 |

Note 1. Writing to these bits is only possible when the RE and TE bits in the SCR are 0 (disabling reception and transmission).

SIMR1 is used to select simple I²C mode and the number of delay stages for the SSDA output.

### IICM Bit (Simple I²C Mode Select)

In conjunction with the SMIF bit in SCMR, this bit selects the operating mode.

### IICDL[4:0] Bits (SSDA Output Delay Select)

These bits are used to set a delay for output on the SSDAn pin relative to the falling edge of the output on the SSCLn pin. The available delay settings range from no delay to 31 cycles, with the clock signal from the on-chip baud rate generator as the base. The signal obtained by frequency-dividing PCLK by the divisor set in SMR.CKS[1:0] is supplied as the clock signal from the on-chip baud rate generator. Set these bits to 00000b unless operation is in simple I²C mode. In simple I²C mode, set the bits to a value in the range from 00001b to 11111b.

## 26.2.13 I²C Mode Register 2 (SIMR2)

Address(es): SCI1.SIMR2 0008 A02Ah, SCI5.SIMR2 0008 A0AAh, SCI12.SIMR2 0008 B30Ah

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | IICACKT | — | — | — | IICCSC | IICINTM |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b0 | IICINTM | I²C Interrupt Mode Select | 0: Use ACK/NACK interrupts.<br>1: Use reception and transmission interrupts. | R/W*1 |
| b1 | IICCSC | Clock Synchronization | 0: No synchronization with the clock signal<br>1: Synchronization with the clock signal | R/W*1 |
| b4 to b2 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |
| b5 | IICACKT | ACK Transmission Data | 0: ACK transmission<br>1: NACK transmission and reception of ACK/NACK | R/W |
| b7, b6 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |

Note 1. Writing to these bits is only possible when the RE and TE bits in the SCR are 0 (serial reception and transmission disabled).

SIMR2 is used to select how reception and transmission are controlled in simple I²C mode.

### IICINTM Bit (I²C Interrupt Mode Select)

This bit selects the sources of interrupt requests in simple I²C mode.

### IICCSC Bit (Clock Synchronization)

Set the IICCSC bit to 1 if the internally generated SSCLn clock signal is to be synchronized when the SSCLn pin has been placed at the low level in the case of a wait inserted by the other device, etc.
The SSCLn clock signal is not synchronized if the IICCSC bit is 0. The SSCLn clock signal is generated in accord with the rate selected in the BRR regardless of the level being input on the SSCLn pin.
Set the IICCSC bit to 1 except during debugging.

### IICACKT Bit (ACK Transmission Data)

Transmitted data contains ACK bits. Set this bit to 1 when ACK and NACK bits are received.

## 26.2.14 I²C Mode Register 3 (SIMR3)

Address(es): SCI1.SIMR3 0008 A02Bh, SCI5.SIMR3 0008 A0ABh, SCI12.SIMR3 0008 B30Bh

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | IICSCLS[1:0] | | IICSDAS[1:0] | | IICSTIF | IICSTP REQ | IICRST AREQ | IICSTA REQ |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b0 | IICSTAREQ | Start Condition Generation | 0: A start condition is not generated.<br>1: A start condition is generated.*1, *3, *4, *5 | R/W |
| b1 | IICRSTAREQ | Restart Condition Generation | 0: A restart condition is not generated.<br>1: A restart condition is generated.*2, *3, *4, *5 | R/W |
| b2 | IICSTPREQ | Stop Condition Generation | 0: A stop condition is not generated.<br>1: A stop condition is generated.*2, *3, *4, *5 | R/W |
| b3 | IICSTIF | Issuing of Start, Restart, or Stop Condition Completed Flag | 0: There are no requests for generating conditions or a condition is being generated.<br>1: A start, restart, or stop condition is completely generated. | R/W |
| b5, b4 | IICSDAS[1:0] | SSDA Output Select | b5 b4<br>0 0: Serial data output<br>0 1: Generate a start, restart, or stop condition.<br>1 0: Output the low level on the SSDAn pin.<br>1 1: Place the SSDAn pin in the high-impedance state. | R/W |
| b7, b6 | IICSCLS[1:0] | SSCL Output Select | b7 b6<br>0 0: Serial clock output<br>0 1: Generate a start, restart, or stop condition.<br>1 0: Output the low level on the SSCLn pin.<br>1 1: Place the SSCLn pin in the high-impedance state. | R/W |

Note 1. Only generate a start condition after checking the bus state and confirming that it is free.
Note 2. Generate a restart or stop condition after checking the bus state and confirming that it is busy.
Note 3. Do not set more than one from among the IICSTAREQ, IICRSTAREQ, and IICSTPREQ bits to 1 at a given time.
Note 4. Execute the generation of a condition after the value of the IICSTIF flag is 0.
Note 5. Do not write 0 to this bit while it is 1. Generation of a condition is suspended by writing 0 to this bit while it is 1.

SIMR3 is used to control the simple I²C mode start and stop conditions, and to hold the SSDAn and SSCLn pins at fixed levels.

### IICSTAREQ Bit (Start Condition Generation)

When a start condition is to be generated, set both the IICSDAS[1:0] and IICSCLS[1:0] bits to 01b as well as setting the IICSTAREQ bit to 1.

[Setting condition]

- Writing 1 to the bit

[Clearing condition]

- Completion of generation of the start condition

#### IICRSTAREQ Bit (Restart Condition Generation)

When a restart condition is to be generated, set both the IICSDAS[1:0] and IICSCLS[1:0] bits to 01b as well as setting the IICRSTAREQ bit to 1.

[Setting condition]

- Writing 1 to the bit

[Clearing condition]

- Completion of generation of the restart condition

#### IICSTPREQ Bit (Stop Condition Generation)

When a stop condition is to be generated, set both the IICSDAS[1:0] and IICSCLS[1:0] bits to 01b as well as setting the IICSTPREQ bit to 1.

[Setting condition]

- Writing 1 to the bit

[Clearing condition]

- Completion of generation of the stop condition

#### IICSTIF Flag (Issuing of Start, Restart, or Stop Condition Completed Flag)

After generating a condition, this bit indicates that the generation is completed. When using the IICSTAREQ, IICRSTAREQ, or IICSTPREQ bit to cause generation of a condition, do so after clearing the IICSTIF flag to 0.

When the IICSTIF flag is 1 while an interrupt request is enabled by setting the SCR.TEIE bit, an STI request is output.

[Setting condition]

- Completion of the generation of a start, restart, or stop condition (however, in cases where this conflicts with any of the conditions for the flag becoming 0 listed below, the other condition takes precedence)

[Clearing conditions]

- Writing 0 to the bit (confirm that the IICSTIF flag is 0 before doing so)
- Writing 0 to the IICM bit in SIMR1 (when operation is not in simple $I^2C$ mode)
- Writing 0 to the TE bit in SCR

#### IICSDAS[1:0] Bits (SSDA Output Select)

These bits control output from the SSDAn pin.

Set the IICSDAS and IICSCLS bits to the same value during normal operations.

#### IICSCLS[1:0] Bits (SSCL Output Select)

These bits control output from the SSCLn pin.

Set the IICSCLS and IICSDAS bits to the same value during normal operations.

## 26.2.15 I²C Status Register (SISR)

Address(es): SCI1.SISR 0008 A02Ch, SCI5.SISR 0008 A0ACh, SCI12.SISR 0008 B30Ch

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | IICACKR |
| Value after reset: | 0 | 0 | x | x | 0 | x | 0 | 0 |

x: Undefined

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b0 | IICACKR | ACK Reception Data Flag | 0: ACK received<br>1: NACK received | R/W*1 |
| b1 | — | Reserved | This bit is read as 0. The write value should be 0. | R/W |
| b2 | — | Reserved | The read value is undefined | R |
| b3 | — | Reserved | This bit is read as 0. The write value should be 0. | R/W |
| b5, b4 | — | Reserved | The read value is undefined | R |
| b7, b6 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |

Note 1. Only 0 can be written to this bit, to clear the flag.

SISR is used to monitor state in relation to simple I²C mode.

### IICACKR Flag (ACK Reception Data Flag)

Received ACK and NACK bits can be read from this bit.

The IICACKR flag is updated at the rising of SSCLn clock for the ACK/NACK receiving bit.

## 26.2.16 SPI Mode Register (SPMR)

Address(es): SCI1.SPMR 0008 A02Dh, SCI5.SPMR 0008 A0ADh, SCI12.SPMR 0008 B30Dh

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | CKPH | CKPOL | — | MFF | — | MSS | CTSE | SSE |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b0 | SSE | SSn# Pin Function Enable | 0: SSn# pin function is disabled.<br>1: SSn# pin function is enabled. | R/W*1 |
| b1 | CTSE | CTS Enable | 0: CTS function is disabled (RTS output function is enabled).<br>1: CTS function is enabled | R/W*1 |
| b2 | MSS | Master Slave Select | 0: Transmission is through the TXDn pin and reception is through the RXDn pin (master mode).<br>1: Reception is through the TXDn pin and transmission is through the RXDn pin (slave mode). | R/W*1 |
| b3 | — | Reserved | This bit is read as 0. The write value should be 0. | R/W |
| b4 | MFF | Mode Fault Flag | 0: No mode-fault error<br>1: Mode-fault error | R/W*2 |
| b5 | — | Reserved | This bit is read as 0. The write value should be 0. | R/W |
| b6 | CKPOL | Clock Polarity Select | 0: Clock polarity is not inverted.<br>1: Clock polarity is inverted. | R/W*1 |
| b7 | CKPH | Clock Phase Select | 0: Clock is not delayed.<br>1: Clock is delayed. | R/W*1 |

Note 1. Writing to these bits is only possible when the RE and TE bits in the SCR are 0 (disabling reception and transmission).
Note 2. Only 0 can be written to these bits, which clears the flag.

SPMR is used to select the extension settings in asynchronous and clock synchronous modes.

### SSE Bit (SSn# Pin Function Enable)

Set this bit to 1 if the SSn# pin is to be used in control of transmission and reception (in simple SPI mode). Set this bit to 0 in any other mode. Furthermore, even for usage in simple SPI mode, the SSn# pin on the master side is not required to control reception and transmission when master mode (SCR.CKE[1:0] = 00b and MSS = 0) is selected and there is a single master, so the setting for the SSE bit is 0. Do not set both the SSE and CTSE bits to enabled (even if this setting is made, operation is the same as that when these bits are cleared to 0).

### CTSE Bit (CTS Enable)

Set this bit to 1 if the SSn# pin is to be used for inputting of the CTS control signal to control of transmission and reception. The RTS signal is output when this bit is set to 0. Set this bit to 0 in smart card interface mode, simple SPI mode, and simple $I^2C$ mode. Do not set both the CTSE and SSE bits to “enabled” (even if this setting is made, operation is the same as that when these bits are cleared to 0).

#### MSS Bit (Master Slave Select)

This bit selects between master and slave operation in simple SPI mode. The functions of the TXDn and RXDn pins are reversed when the MSS bit is set to 1, so that data are received through the TXDn pin and transmitted through the RXDn pin.
Set this bit to 0 in modes other than simple SPI mode.

#### MFF Flag (Mode Fault Flag)

This bit indicates mode-fault errors.
In a multi-master configuration, determine the mode-fault error occurrence by reading the MFF flag.
[Setting condition]

- Input on the SSn# pin being at the low level during master operation in simple SPI mode (SSE bit = 1 and MSS bit = 0)

[Clearing condition]

- Writing 0 to the bit after it was read as 1

#### CKPOL Bit (Clock Polarity Select)

This bit selects the polarity of the clock signal output through the SCKn pin. See Figure 26.56 for details.
Clear the bit to 0 in other than simple SPI mode and clock synchronous mode.

#### CKPH Bit (Clock Phase Select)

This bit selects the phase of the clock signal output through the SCKn pin. See Figure 26.56 for details.
Clear the bit to 0 in other than simple SPI mode and clock synchronous mode.

### 26.2.17 Extended Serial Module Enable Register (ESMER)

Address(es): SCI12.ESMER 0008 B320h

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | ESME |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b0 | ESME | Extended Serial Mode Enable | 0: The extended serial mode is disabled.<br>1: The extended serial mode is enabled. | R/W |
| b7 to b1 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |

#### ESME Bit (Extended Serial Mode Enable)

When the ESME bit is 1, the facilities of the extended serial mode control section are enabled.
When the ESME bit is 0, the extended serial mode control section enters the following states:

- the extended serial mode control section is initialized

**Table 26.23 Settings of the ESME Bit and Guaranteed Operation by Timer Operation Mode**

| ESME Bit | Timer Mode | Break Field Low Width Determination Mode | Break Field Low Width Output Mode |
|---|---|---|---|
| 0 | ○*1 | × | × |
| 1 | ○ | ○ | ○ |

○:Guarantee of operation is necessary. ×:Guarantee of operation is not necessary.
Note 1. Operation is only possible with PCLK selected.

## 26.2.18 Control Register 0 (CR0)

Address(es): SCI12.CR0 0008 B321h

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | BRME | RXDSF | SFSF | — |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b0 | — | Reserved | This bit is read as 0. The write value should be 0. | R/W |
| b1 | SFSF | Start Frame Status Flag | 0: Start Frame detection function is disabled.<br>1: Start Frame detection function is enabled. | R |
| b2 | RXDSF | RXDX12 Input Status Flag | 0: RXDX12 input is enabled.<br>1: RXDX12 input is disabled. | R |
| b3 | BRME | Bit Rate Measurement Enable | 0: Measurement of bit rate is disabled.<br>1: Measurement of bit rate is enabled. | R/W |
| b7 to b4 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |

## 26.2.19 Control Register 1 (CR1)

Address(es): SCI12.CR1 0008 B322h

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | PIBS[2:0] | | | PIBE | CF1DS[1:0] | | CF0RE | BFE |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b0 | BFE | Break Field Enable | 0: Break Field detection is disabled.<br>1: Break Field detection is enabled. | R/W |
| b1 | CF0RE | Control Field 0 Reception Enable | 0: Reception of Control Field 0 is disabled.<br>1: Reception of Control Field 0 is enabled. | R/W |
| b3, b2 | CF1DS[1:0] | Control Field 1 Data Register Select | b3 b2<br>0 0: Selects comparison with the value in PCF1DR.<br>0 1: Selects comparison with the value in SCF1DR.<br>1 0: Selects comparison with the values in PCF1DR and SCF1DR.<br>1 1: Setting prohibited. | R/W |
| b4 | PIBE | Priority Interrupt Bit Enable | 0: The priority interrupt bit is disabled.<br>1: The priority interrupt bit is enabled. | R/W |
| b7 to b5 | PIBS[2:0] | Priority Interrupt Bit Select | b7 b5<br>0 0 0: 0th bit of Control Field 1<br>0 0 1: 1st bit of Control Field 1<br>0 1 0: 2nd bit of Control Field 1<br>0 1 1: 3rd bit of Control Field 1<br>1 0 0: 4th bit of Control Field 1<br>1 0 1: 5th bit of Control Field 1<br>1 1 0: 6th bit of Control Field 1<br>1 1 1: 7th bit of Control Field 1 | R/W |

## 26.2.20 Control Register 2 (CR2)

Address(es): SCI12.CR2 0008 B320h

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | RTS[1:0] | | BCCS[1:0] | | — | DFCS[2:0] | | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b2 to b0 | DFCS[2:0] | RXDX12 Signal Digital Filter Clock Select | b2 b0<br>0 0 0: Filter is disabled.<br>0 0 1: Filter clock is SCI base clock*1<br>0 1 0: Filter clock is PCLK/8<br>0 1 1: Filter clock is PCLK/16<br>1 0 0: Filter clock is PCLK/32<br>1 0 1: Filter clock is PCLK/64<br>1 1 0: Filter clock is PCLK/128<br>1 1 1: Setting prohibited | R/W |
| b3 | — | Reserved | This bit is read as 0. The write value should be 0. | R/W |
| b5, b4 | BCCS[1:0] | Bus Collision Detection Clock Select | b5 b4<br>0 0: SCI base clock<br>0 1: SCI base clock frequency divided by 2<br>1 0: SCI base clock frequency divided by 4<br>1 1: Setting prohibited | R/W |
| b7, b6 | RTS[1:0] | RXDX12 Reception Sampling Timing Select | • When SCI12.SEMR.ABCS = 0<br>b7 b6<br>0 0: Rising edge of the 8th cycle of SCI base clock<br>0 1: Rising edge of the 10th cycle of SCI base clock<br>1 0: Rising edge of the 12th cycle of SCI base clock<br>1 1: Rising edge of the 14th cycle of SCI base clock<br><br>• When SCI12.SEMR.ABCS = 1<br>b7 b6<br>0 0: Rising edge of the 4th cycle of SCI base clock<br>0 1: Rising edge of the 5th cycle of SCI base clock<br>1 0: Rising edge of the 6th cycle of SCI base clock<br>1 1: Rising edge of the 7th cycle of SCI base clock | R/W |

Note: • The period of the SCI base clock is 1/16 of a single bit period when the SCI12.SEMR.ABCS is 0, and 1/8 of a single bit period when the SCI12.SEMR.ABCS is 1.

Note 1. To use the SCI base clock, set the SCI12.SCR.TE bit to 1.

## 26.2.21 Control Register 3 (CR3)

Address(es): SCI12.CR3 0008 B324h

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | SDST |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b0 | SDST | Start Frame Detection Start | 0: Detection of Start Frame is not performed.<br>1: Detection of Start Frame is performed. | R/W |
| b7 to b1 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |

### SDST Bit (Start Frame Detection Start)

Detection of a Start Frame begins when this bit is set to 1. The bit is read as 0.

## 26.2.22 Port Control Register (PCR)

Address(es): SCI12.PCR 0008 B325h

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | — | SHARPS | — | — | RXDXPS | TXDXPS |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b0 | TXDXPS | TXDX12 Signal Polarity Select | 0: The polarity of TXDX12 signal is not inverted for output.<br>1: The polarity of TXDX12 signal is inverted for output. | R/W |
| b1 | RXDXPS | RXDX12 Signal Polarity Select | 0: The polarity of RXDX12 signal is not inverted for input.<br>1: The polarity of RXDX12 signal is inverted for input. | R/W |
| b3, b2 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |
| b4 | SHARPS | TXDX12/RXDX12 Pin Multiplexing Select | 0: The TXDX12 and RXDX12 pins are independent.<br>1: The TXDX12 and RXDX12 signals are multiplexed on the same pin. | R/W |
| b7 to b5 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |

### SHARPS Bit (TXDX12/RXDX12 Pin Multiplexing Select)

When this bit is set to 1, the TXDX12 and RXDX12 signals are multiplexed on the same pin so that half-duplex communications become possible.

## 26.2.23 Interrupt Control Register (ICR)

Address(es): SCI12.ICR 0008 B326h

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | AEDIE | BCDIE | PIBDIE | CF1MIE | CF0MIE | BFDIE |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b0 | BFDIE | Break Field Low Width Detected Interrupt Enable | 0: Interrupts on detection of the low width for a Break Field are disabled.<br>1: Interrupts on detection of the low width for a Break Field are enabled. | R/W |
| b1 | CF0MIE | Control Field 0 Match Detected Interrupt Enable | 0: Interrupts on detection of a match with Control Field 0 are disabled.<br>1: Interrupts on detection of a match with Control Field 0 are enabled. | R/W |
| b2 | CF1MIE | Control Field 1 Match Detected Interrupt Enable | 0: Interrupts on detection of a match with Control Field 1 are disabled.<br>1: Interrupts on detection of a match with Control Field 1 are enabled. | R/W |
| b3 | PIBDIE | Priority Interrupt Bit Detected Interrupt Enable | 0: Interrupts on detection of the priority interrupt bit are disabled.<br>1: Interrupts on detection of the priority interrupt bit are enabled. | R/W |
| b4 | BCDIE | Bus Collision Detected Interrupt Enable | 0: Interrupts on detection of a bus collision are disabled.<br>1: Interrupts on detection of a bus collision are enabled. | R/W |
| b5 | AEDIE | Valid Edge Detected Interrupt Enable | 0: Interrupts on detection of a valid edge are disabled.<br>1: Interrupts on detection of a valid edge are enabled. | R/W |
| b7, b6 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |

## 26.2.24 Status Register (STR)

Address(es): SCI12.STR 0008 B327h

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | AEDF | BCDF | PIBDF | CF1MF | CF0MF | BFDF |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b0 | BFDF | Break Field Low Width Detection Flag | [Setting conditions]<br>• Detection of the low width for a Break Field<br>• Completion of the output of the low width for a Break Field<br>• Underflow of the timer<br>[Clearing condition]<br>• Writing 1 to the BFDCL bit in STCR | R |
| b1 | CF0MF | Control Field 0 Match Flag | [Setting condition]<br>• A match between the value received in Control Field 0 and the set value.<br>[Clearing condition]<br>• Writing 1 to the CF0MCL bit in STCR | R |
| b2 | CF1MF | Control Field 1 Match Flag | [Setting condition]<br>• A match between the data received in Control Field 1 and the set values.<br>[Clearing condition]<br>• Writing 1 to the CF1MCL bit in STCR | R |
| b3 | PIBDF | Priority Interrupt Bit Detection Flag | [Setting condition]<br>• Detection of the priority interrupt bit<br>[Clearing condition]<br>• Writing 1 to the PIBDCL bit in STCR | R |
| b4 | BCDF | Bus Collision Detected Flag | [Setting condition]<br>• Detection of the bus collision<br>[Clearing condition]<br>• Writing 1 to the BCDCL bit in STCR | R |
| b5 | AEDF | Valid Edge Detection Flag | [Setting condition]<br>• Detection of a valid edge<br>[Clearing condition]<br>• Writing 1 to the AEDCL bit in STCR | R |
| b7, b6 | — | Reserved | These bits are read as 0. The write value should be 0. | R |

## 26.2.25 Status Clear Register (STCR)

Address(es): SCI12.STCR 0008 B328h

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | AEDCL | BCDCL | PIBDCL | CF1MCL | CF0MCL | BFDCL |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b0 | BFDCL | BFDF Clear | Setting this bit to 1 clears the STR.BFDF flag. This bit is read as 0. | R/W |
| b1 | CF0MCL | CF0MF Clear | Setting this bit to 1 clears the STR.CF0MF flag. This bit is read as 0. | R/W |
| b2 | CF1MCL | CF1MF Clear | Setting this bit to 1 clears the STR.CF1MF flag. This bit is read as 0. | R/W |
| b3 | PIBDCL | PIBDF Clear | Setting this bit to 1 clears the STR.PIBDF flag. This bit is read as 0. | R/W |
| b4 | BCDCL | BCDF Clear | Setting this bit to 1 clears the STR.BCDF flag. This bit is read as 0. | R/W |
| b5 | AEDCL | AEDF Clear | Setting this bit to 1 clears the STR.AEDF flag. This bit is read as 0. | R/W |
| b7, b6 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |

## 26.2.26 Control Field 0 Data Register (CF0DR)

Address(es): SCI12.CF0DR 0008 B329h

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

CF0DR is an 8-bit readable and writable register that holds a value for comparison with Control Field 0.

## 26.2.27 Control Field 0 Compare Enable Register (CF0CR)

Address(es): SCI12.CF0CR 0008 B32Ah

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | CF0CE7 | CF0CE6 | CF0CE5 | CF0CE4 | CF0CE3 | CF0CE2 | CF0CE1 | CF0CE0 |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b0 | CF0CE0 | Control Field 0 Bit 0 Compare Enable | 0: Comparison with bit 0 of Control Field 0 is disabled.<br>1: Comparison with bit 0 of Control Field 0 is enabled. | R/W |
| b1 | CF0CE1 | Control Field 0 Bit 1 Compare Enable | 0: Comparison with bit 1 of Control Field 0 is disabled.<br>1: Comparison with bit 1 of Control Field 0 is enabled. | R/W |
| b2 | CF0CE2 | Control Field 0 Bit 2 Compare Enable | 0: Comparison with bit 2 of Control Field 0 is disabled.<br>1: Comparison with bit 2 of Control Field 0 is enabled. | R/W |
| b3 | CF0CE3 | Control Field 0 Bit 3 Compare Enable | 0: Comparison with bit 3 of Control Field 0 is disabled.<br>1: Comparison with bit 3 of Control Field 0 is enabled. | R/W |
| b4 | CF0CE4 | Control Field 0 Bit 4 Compare Enable | 0: Comparison with bit 4 of Control Field 0 is disabled.<br>1: Comparison with bit 4 of Control Field 0 is enabled. | R/W |
| b5 | CF0CE5 | Control Field 0 Bit 5 Compare Enable | 0: Comparison with bit 5 of Control Field 0 is disabled.<br>1: Comparison with bit 5 of Control Field 0 is enabled. | R/W |
| b6 | CF0CE6 | Control Field 0 Bit 6 Compare Enable | 0: Comparison with bit 6 of Control Field 0 is disabled.<br>1: Comparison with bit 6 of Control Field 0 is enabled. | R/W |
| b7 | CF0CE7 | Control Field 0 Bit 7 Compare Enable | 0: Comparison with bit 7 of Control Field 0 is disabled.<br>1: Comparison with bit 7 of Control Field 0 is enabled. | R/W |

## 26.2.28 Control Field 0 Receive Data Register (CF0RR)

Address(es): SCI12.CF0RR 0008 B32Bh

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

CF0RR is a read-only register that holds the value received in Control Field 0. Writing to this register from the CPU or DTC is not possible.

## 26.2.29 Primary Control Field 1 Data Register (PCF1DR)

Address(es): SCI12.PCF1DR 0008 B32Ch

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

PCF1DR is an 8-bit readable and writable register that holds the 8-bit primary value for comparison with Control Field 1.

## 26.2.30 Secondary Control Field 1 Data Register (SCF1DR)

Address(es): SCI12.SCF1DR 0008 B32Dh

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

PCF1DR is an 8-bit readable and writable register that holds the 8-bit secondary value for comparison with Control Field 1.

## 26.2.31 Control Field 1 Compare Enable Register (CF1CR)

Address(es): SCI12.CF1CR 0008 B32Eh

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | CF1CE7 | CF1CE6 | CF1CE5 | CF1CE4 | CF1CE3 | CF1CE2 | CF1CE1 | CF1CE0 |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b0 | CF1CE0 | Control Field 1 Bit 0 Compare Enable | 0: Comparison with bit 0 of Control Field 1 is disabled.<br>1: Comparison with bit 0 of Control Field 1 is enabled. | R/W |
| b1 | CF1CE1 | Control Field 1 Bit 1 Compare Enable | 0: Comparison with bit 1 of Control Field 1 is disabled.<br>1: Comparison with bit 1 of Control Field 1 is enabled. | R/W |
| b2 | CF1CE2 | Control Field 1 Bit 2 Compare Enable | 0: Comparison with bit 2 of Control Field 1 is disabled.<br>1: Comparison with bit 2 of Control Field 1 is enabled. | R/W |
| b3 | CF1CE3 | Control Field 1 Bit 3 Compare Enable | 0: Comparison with bit 3 of Control Field 1 is disabled.<br>1: Comparison with bit 3 of Control Field 1 is enabled. | R/W |
| b4 | CF1CE4 | Control Field 1 Bit 4 Compare Enable | 0: Comparison with bit 4 of Control Field 1 is disabled.<br>1: Comparison with bit 4 of Control Field 1 is enabled. | R/W |
| b5 | CF1CE5 | Control Field 1 Bit 5 Compare Enable | 0: Comparison with bit 5 of Control Field 1 is disabled.<br>1: Comparison with bit 5 of Control Field 1 is enabled. | R/W |
| b6 | CF1CE6 | Control Field 1 Bit 6 Compare Enable | 0: Comparison with bit 6 of Control Field 1 is disabled.<br>1: Comparison with bit 6 of Control Field 1 is enabled. | R/W |
| b6 | CF1CE7 | Control Field 1 Bit 7 Compare Enable | 0: Comparison with bit 7 of Control Field 1 is disabled.<br>1: Comparison with bit 7 of Control Field 1 is enabled. | R/W |

## 26.2.32 Control Field 1 Receive Data Register (CF1RR)

Address(es): SCI12.CF1RR 0008 B32Fh

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

CF1RR is a read-only register that holds the value received in Control Field 1. Writing to this register from the CPU or DTC is not possible.

## 26.2.33 Timer Control Register (TCR)

Address(es): SCI12.TCR 0008 B330h

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | TCST |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b0 | TCST | Timer Count Start | 0: Stops the timer counting<br>1: Starts the timer counting | R/W |
| b7 to b1 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |

## 26.2.34 Timer Mode Register (TMR)

Address(es): SCI12.TMR 0008 B331h

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | TCSS[2:0] | | | TWRC | — | TOMS[1:0] | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b1, b0 | TOMS[1:0] | Timer Operating Mode Select*1 | b1 b0<br>0 0: Timer mode<br>0 1: Break Field low width determination mode<br>1 0: Break Field low width output mode<br>1 1: Setting prohibited | R/W |
| b2 | — | Reserved | This bit is read as 0. The write value should be 0. | R/W |
| b3 | TWRC | Counter Write Control | 0: Data is written to the reload register and counter<br>1: Data is written to the reload register only | R/W |
| b6 to b4 | TCSS[2:0] | Timer Count Clock Source Select*1 | b6 b4<br>0 0 0: PCLK<br>0 0 1: PCLK/2<br>0 1 0: PCLK/4<br>0 1 1: PCLK/8<br>1 0 0: PCLK/16<br>1 0 1: PCLK/32<br>1 1 0: PCLK/64<br>1 1 1: PCLK/128 | R/W |
| b7 | — | Reserved | This bit is read as 0. The write value should be 0. | R/W |

Note 1. Rewrite the TOMS[1:0] and TCSS[2:0] bits only when the timer is stopped (TCST = 0).

### TWRC Bit (Counter Write Control)

This bit determines whether a value written to TPRE or TCNT is written to the reload register only or is written to both the reload register and the counter.

## 26.2.35 Timer Prescaler Register (TPRE)

Address(es): SCI12.TPRE 0008 B332h

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| Value after reset: | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

TPRE consists of an 8-bit reload register, a read buffer, and a counter, each of which has FFh as its initial value. The counter counts down in synchronization with the counter clock selected by the TMR.TCSS[2:0] bits, and is reloaded with the value in the reload register when it underflows. Underflows of this register provide the clock source to drive counting by the TCNT register. The reload register and read buffer are allocated to the same address, so in writing, transfer is to the reload register and in reading, transfer is of the counter value from the read buffer.
It takes one PCLK cycle to load a value from the reload register to the counter.

## 26.2.36 Timer Count Register (TCNT)

Address(es): SCI12.TCNT 0008 B333h

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| Value after reset: | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

TCNT consists of an 8-bit reload register, a read buffer, and a counter, each of which has FFh as its initial value. This down-counter counts underflows of the TPRE register until the TCNT register underflows, and is then reloaded with the value from the reload register. The reload register and read buffer are allocated to the same address, so in writing, transfer is to the reload register and in reading, transfer is of the counter value from the read buffer.
It takes one PCLK cycle to load a value from the reload register to the counter.

## 26.3 Operation in Asynchronous Mode

Figure 26.5 shows the general format for asynchronous serial communications.
One frame consists of a start bit (low level), transmit/receive data, a parity bit, and stop bits (high level).
In asynchronous serial communications, the communications line is usually held in the mark state (high level).
The SCI monitors the communications line, and when it goes to the space state (low level), recognizes a start bit and starts serial communications.
Inside the SCI, the transmitter and receiver are independent units, enabling full-duplex communications. Both the transmitter and the receiver also have a double-buffered structure, so that data can be read or written during transmission or reception, enabling continuous data transmission and reception.

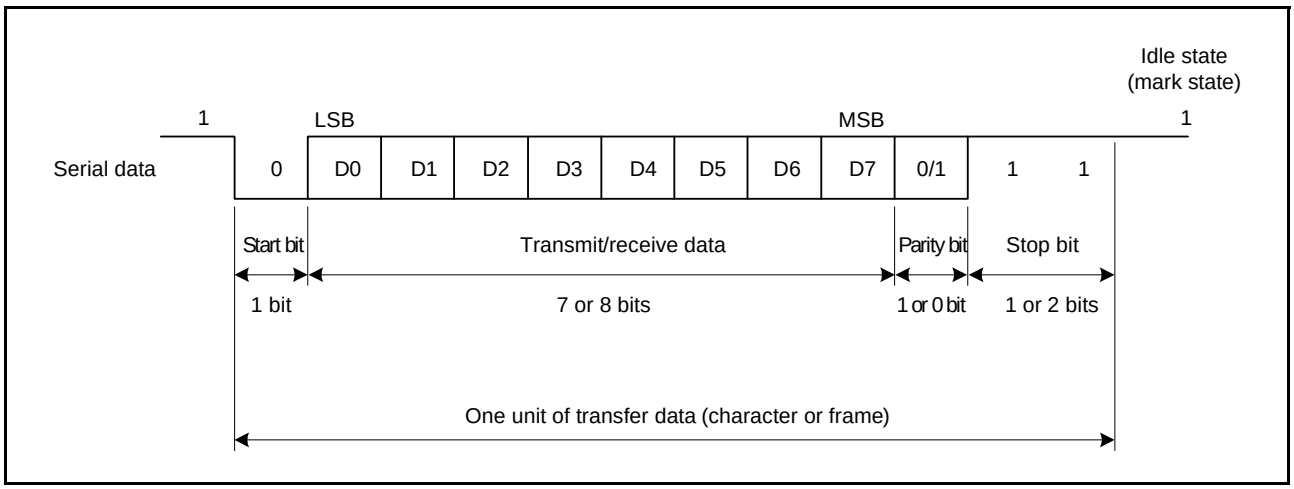



**Figure 26.5 Data Format in Asynchronous Serial Communications (Example with 8-Bit Data, Parity, 2 Stop Bits)**

### 26.3.1 Serial Data Transfer Format

Table 26.24 lists the serial data transfer formats that can be used in asynchronous mode.
Any of 12 transfer formats can be selected according to the SMR setting. For details of multi-processor function, refer to section 26.4, Multi-Processor Communications Function.

**Table 26.24 Serial Transfer Formats (Asynchronous Mode)**

| SMR Setting | | | | Serial Transfer Format and Frame Length |
|---|---|---|---|---|
| CHR | PE | MP | STOP | 1 2 3 4 5 6 7 8 9 10 11 12 |
| 0 | 0 | 0 | 0 | S \| 8-bit data \| STOP |
| 0 | 0 | 0 | 1 | S \| 8-bit data \| STOP \| STOP |
| 0 | 1 | 0 | 0 | S \| 8-bit data \| P \| STOP |
| 0 | 1 | 0 | 1 | S \| 8-bit data \| P \| STOP \| STOP |
| 1 | 0 | 0 | 0 | S \| 7-bit data \| STOP |
| 1 | 0 | 0 | 1 | S \| 7-bit data \| STOP \| STOP |
| 1 | 1 | 0 | 0 | S \| 7-bit data \| P \| STOP |
| 1 | 1 | 0 | 1 | S \| 7-bit data \| P \| STOP \| STOP |
| 0 | — | 1 | 0 | S \| 8-bit data \| MPB \| STOP |
| 0 | — | 1 | 1 | S \| 8-bit data \| MPB \| STOP \| STOP |
| 1 | — | 1 | 0 | S \| 7-bit data \| MPB \| STOP |
| 1 | — | 1 | 1 | S \| 7-bit data \| MPB \| STOP \| STOP |

S: Start bit
STOP: Stop bit
P: Parity bit
MPB: Multi-processor bit

### 26.3.2 Receive Data Sampling Timing and Reception Margin in Asynchronous Mode

In asynchronous mode, the SCI operates on a base clock with a frequency of 16 times*1 the bit rate.
In reception, the SCI samples the falling edge of the start bit using the base clock, and performs internal synchronization. Since receive data is sampled at the rising edge of the 8th pulse*1 of the base clock, data is latched at the middle of each bit, as shown in Figure 26.6. Thus the reception margin in asynchronous mode is determined by formula (1) below.

$$M = \left| (0.5 - \frac{1}{2N}) - (L - 0.5)\ F - \frac{D - 0.5}{N}\ (1 + F) \right| \times 100\ [\%] \cdots \text{Formula (1)}$$

M: Reception margin
N: Ratio of bit rate to clock (N = 16 when ABCS in SEMR = 0, N = 8 when ABCS in SEMR = 1)
D: Duty cycle of clock (D = 0.5 to 1.0)
L: Frame length (L = 9 to 13)
F: Absolute value of clock frequency deviation

Assuming values of F = 0 and D = 0.5 in formula (1), the reception margin is determined by the formula below.

M = {0.5 - 1/(2 × 16)} × 100 (%) = 46.875%

However, this is only the computed value, and a margin of 20% to 30% should be allowed in system design.

Note 1. This is an example when the ABCS bit in SEMR is 0. When the ABCS bit is 1, a frequency of 8 times the bit rate is used as a base clock and receive data is sampled at the rising edge of the 4th pulse of the base clock.

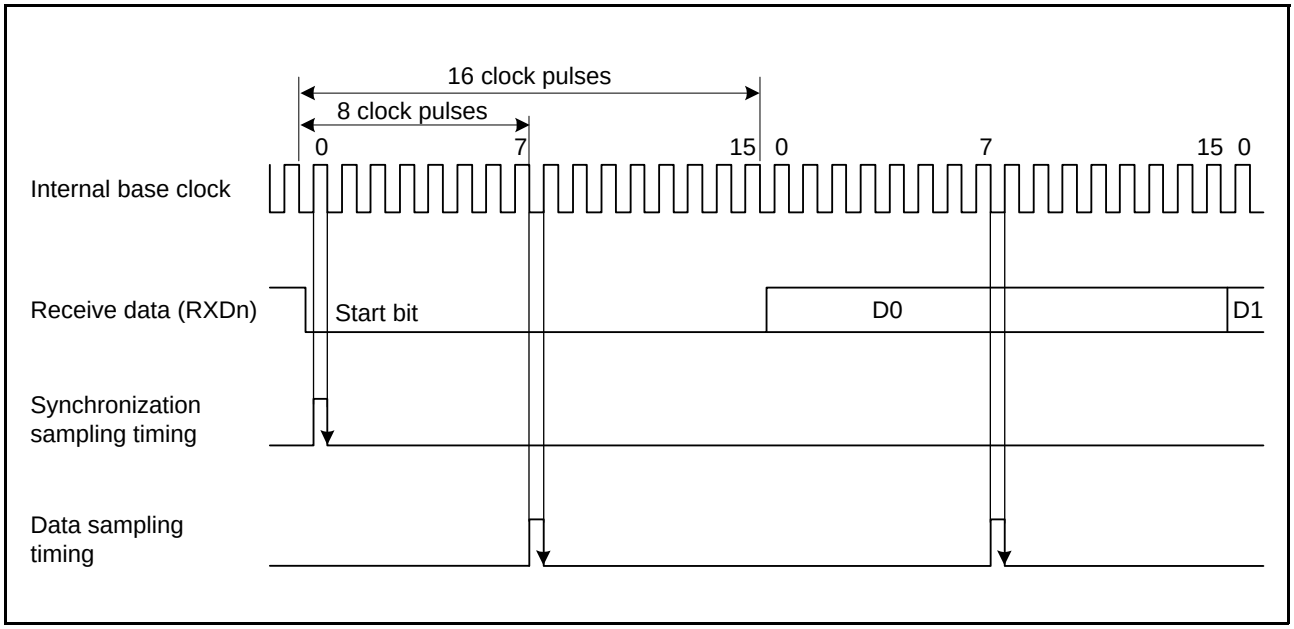



**Figure 26.6 Receive Data Sampling Timing in Asynchronous Mode**

### 26.3.3 Clock

Either an internal clock generated by the on-chip baud rate generator or an external clock input to the SCKn pin can be selected as the SCI's transfer clock, according to the setting of the CM bit in SMR and the CKE[1:0] bits in SCR.
When an external clock is input to the SCKn pin, the clock frequency should be 16 times the bit rate (when ABCS in SEMR = 0) and 8 times the bit rate (when ABCS in SEMR = 1). In addition, when an external clock is specified, the base clock of MTIOC1A and MTIOC2A can be selected by the SCIn.SEMR.ACS0 bit (n = 1, 5, 12).
When the SCI is operated on an internal clock, the clock can be output from the SCKn pin. The frequency of the clock output in this case is equal to the bit rate, and the phase is such that the rising edge of the clock is in the middle of the transmit data, as shown in Figure 26.7.

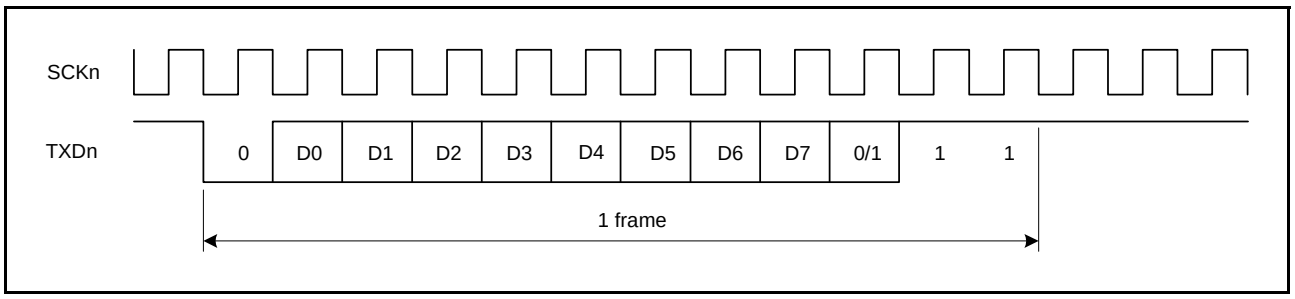



**Figure 26.7 Phase Relationship between Output Clock and Transmit Data (Asynchronous Mode: SMR.CHR = 0, PE = 1, MP = 0, STOP = 1)**

### 26.3.4 CTS and RTS Functions

The CTS function is the use of input on the CTSn# pin in transmission control. Setting the SPMR.CTSE bit to 1 enables the CTS function. When the CTS function is enabled, placing the low level on the CTSn# pin causes transmission to start.
Applying the high level to the CTS# pin while transmission is in progress does not affect transmission of the current frame, which continues.
In the RTS function, by using the function of output on the RTSn# pin, a low level is output when reception becomes possible. Conditions for output of the low and high level are shown below.

[Conditions for low-level output]
Satisfaction of all conditions listed below

- The value of the RE bit in the SCR is 1
- Reception is not in progress
- There are no received data yet to be read
- The ORER, FER, and PER flags in the SSR are all 0

[Condition for high-level output]

- The conditions for low-level output have not been satisfied.

### 26.3.5 SCI Initialization (Asynchronous Mode)

Before transmitting and receiving data, start by writing the initial value 00h to SCR and then continue through the procedure for SCI given in Figure 26.8. Whenever the operating mode or transfer format is changed, SCR must be initialized before the change is made.

When the external clock is used in asynchronous mode, ensure that the clock signal is supplied even during initialization.

Note that clearing the SCR.RE bit to 0 initializes neither the ORER, FER, and PER flags in SSR nor RDR.

Moreover, note that switching the value of the SCR.TE bit from 1 to 0 or 0 to 1 while the SCR.TIE bit is 1 leads to the generation of a TXI interrupt request.

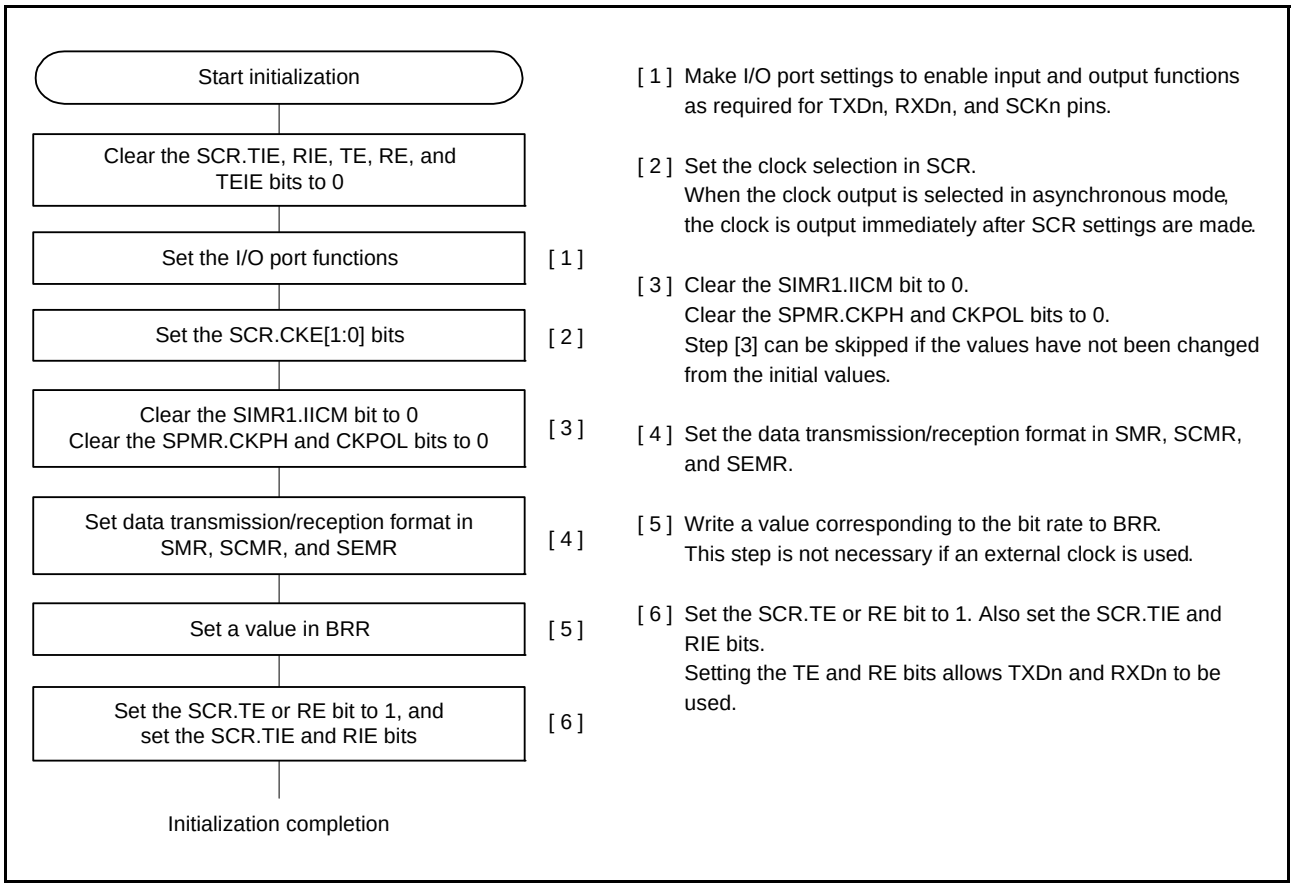


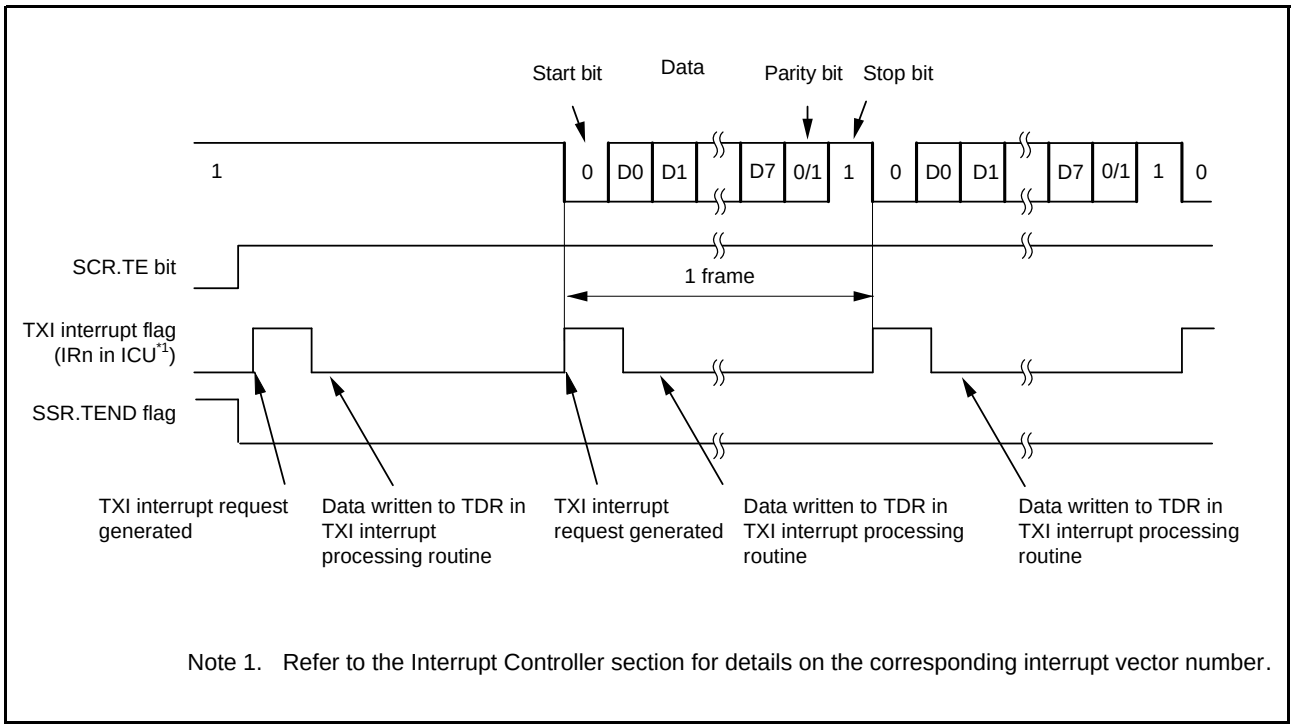

**Figure 26.8 Sample SCI Initialization Flowchart (Asynchronous Mode)**

### 26.3.6 Serial Data Transmission (Asynchronous Mode)

Figure 26.9, Figure 26.10, and Figure 26.11 shows an example of the operation for serial transmission in asynchronous mode.
In serial transmission, the SCI operates as described below.

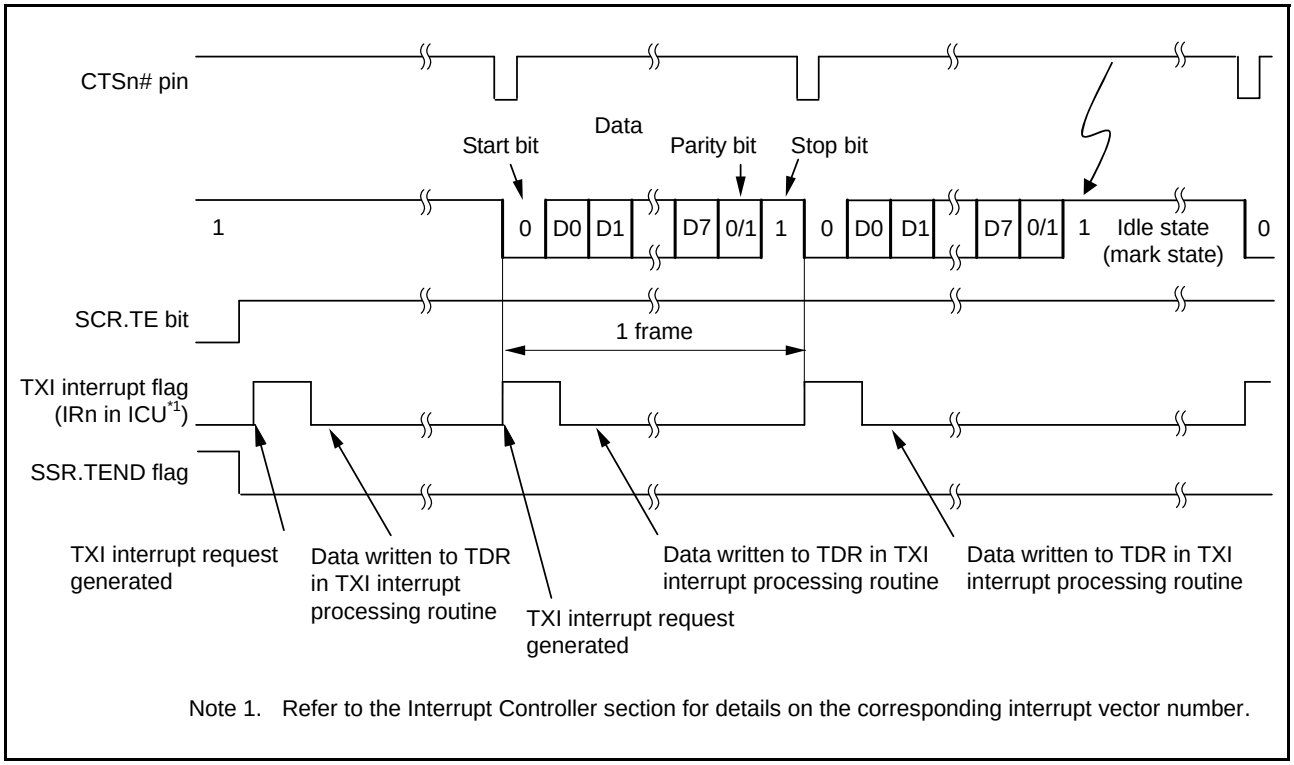
1. The SCI transfers data from TDR to TSR when data is written to TDR in the TXI interrupt processing routine. The TXI interrupt request at the beginning of transmission is generated when the TE bit in SCR is set to 1 after the TIE bit in SCR is set to 1 or when these two bits are set to 1 simultaneously by a single instruction.
2. Transmission starts after the CTSE bit in SPMR is set to 0 (CTS function is disabled) and a low level on the CTSn# pin causes data transfer from TDR to TSR. If the TIE bit in SCR is 1 at this time, a TXI interrupt request is generated. Continuous transmission is obtainable by writing the next transmit data to TDR in the TXI interrupt processing routine before transmission of the current transmit data is completed. When TEI interrupt requests are in use, set the SCR.TIE bit to 0 (a TXI interrupt request is disabled) and the SCR.TEIE bit to 1 (a TEI interrupt request is enabled) after the last of the data to be transmitted are written to the TDR from the processing routine for TXI requests.
3. Data is sent from the TXDn pin in the following order: start bit, transmit data, parity bit or multi-processor bit (may be omitted depending on the format), and stop bit.
4. The SCI checks for updating of (writing to) TDR at the time of stop bit output.
5. When TDR is updated, setting of the CTSE bit in SPMR to 0 (CTS function is disabled) or a low level input on the CTSn# pin cause the next transfer of the next transmit data from TDR to TSR and sending of the stop bit, after which serial transmission of the next frame starts.
6. If TDR is not updated, the TEND flag in SSR is set to 1, the stop bit is sent, and then the mark state is entered in which 1 is output. If the TEIE bit in SCR is 1 at this time, the TEND flag in SSR is set to 1 and a TEI interrupt request is generated.

Figure 26.12 shows a sample flowchart for serial transmission in asynchronous mode.



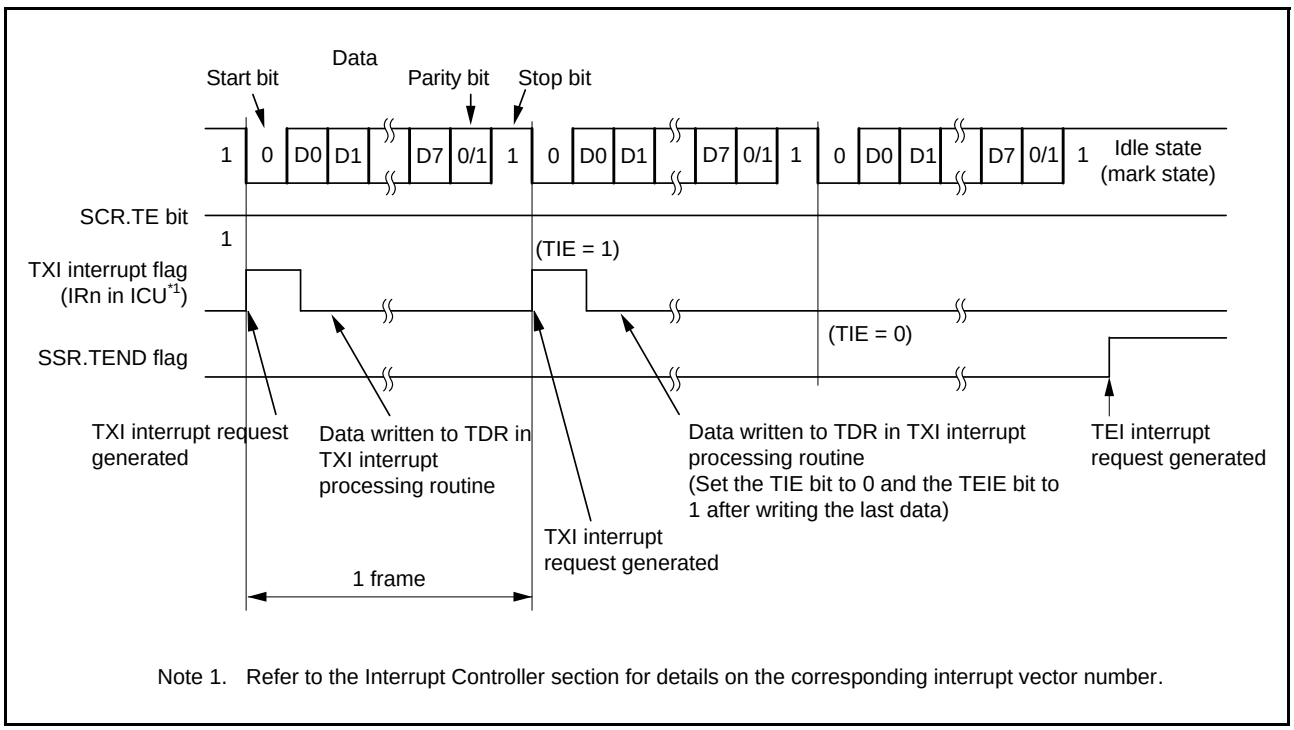

Note 1. Refer to the Interrupt Controller section for details on the corresponding interrupt vector number.

**Figure 26.9 Example of Operation for Serial Transmission in Asynchronous Mode (1)**
**(with 8-Bit Data, Parity, 1 Stop Bit, CTS Function Not Used, at the Beginning of Transmission)**



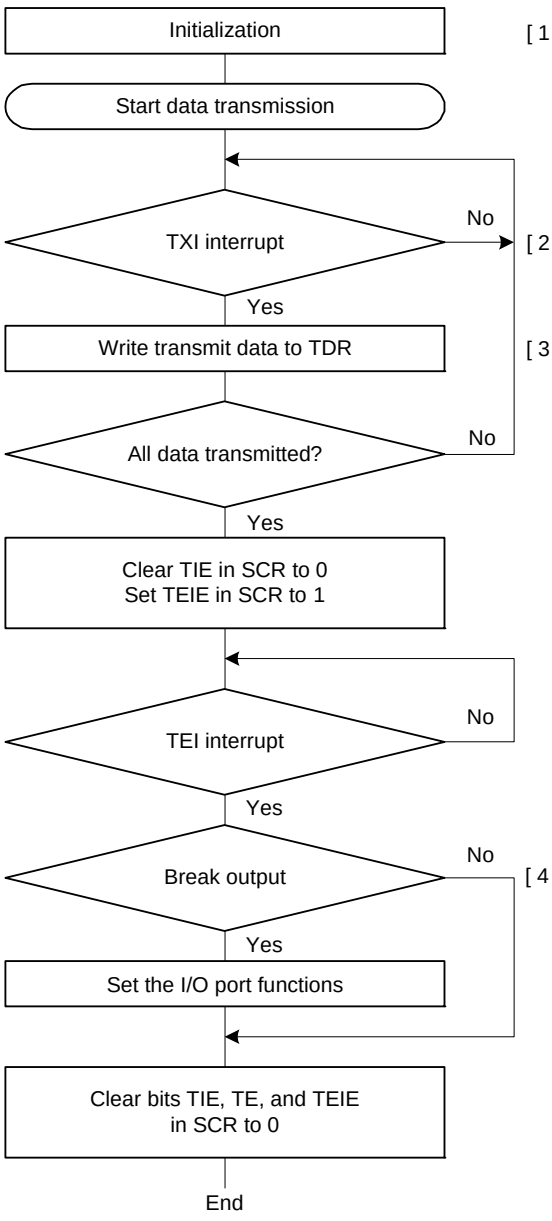

Note 1. Refer to the Interrupt Controller section for details on the corresponding interrupt vector number.

**Figure 26.10 Example of Operation for Serial Transmission in Asynchronous Mode (2) (with 8-Bit Data, Parity, 1 Stop Bit, CTS Function Used, at the Beginning of Transmission)**




Note 1. Refer to the Interrupt Controller section for details on the corresponding interrupt vector number.

**Figure 26.11 Example of Operation for Serial Transmission in Asynchronous Mode (3) (with 8-Bit Data, Parity, 1 Stop Bit, CTS Function Not Used, from the Middle of Transmission until Transmission Completion)**




[ 1 ] SCI initialization:
Set data transmission.
After the TE bit in SCR is set to 1, 1 is output for a frame, and transmission is enabled.

[ 2 ] Transmit data write to TDR by a TXI interrupt request:
When transmit data is transferred from TDR to TSR, a transmit data empty interrupt (TXI) request is generated.
Write transmit data to TDR once in the TXI interrupt processing routine.

[ 3 ] Serial transmission continuation procedure:
To continue serial transmission, write transmit data to TDR once using a TXI interrupt request.
Transmit data can also be written to TDR by activating the DTC.
When TEI interrupt requests are in use, set the SCR.TIE bit to 0 and the SCR.TEIE bit to 1 after the last of the data to be transmitted are written to the TDR.

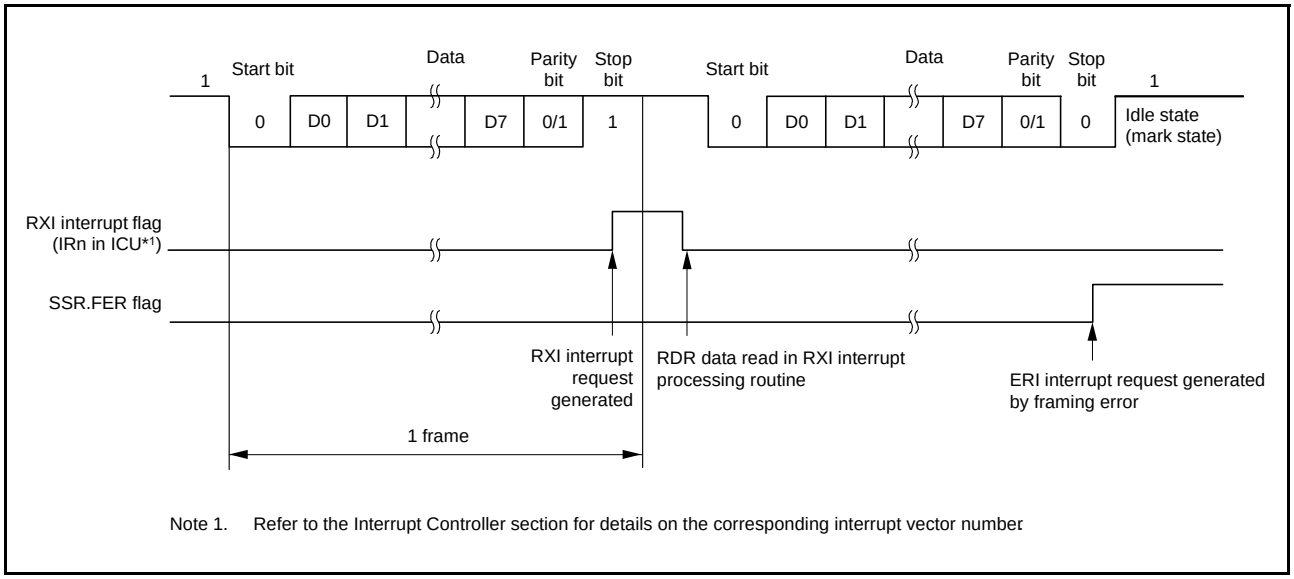
[ 4 ] Break output at the end of serial transmission:
To output a break in serial transmission, set the I/O port function corresponding to the TXDn pin (corresponding to output of the low level), and after switching the TXDn pin to the general I/O port function, set the TE bit in the SCR to 0.

**Figure 26.12 Example of Serial Transmission Flowchart in Asynchronous Mode**

### 26.3.7 Serial Data Reception (Asynchronous Mode)

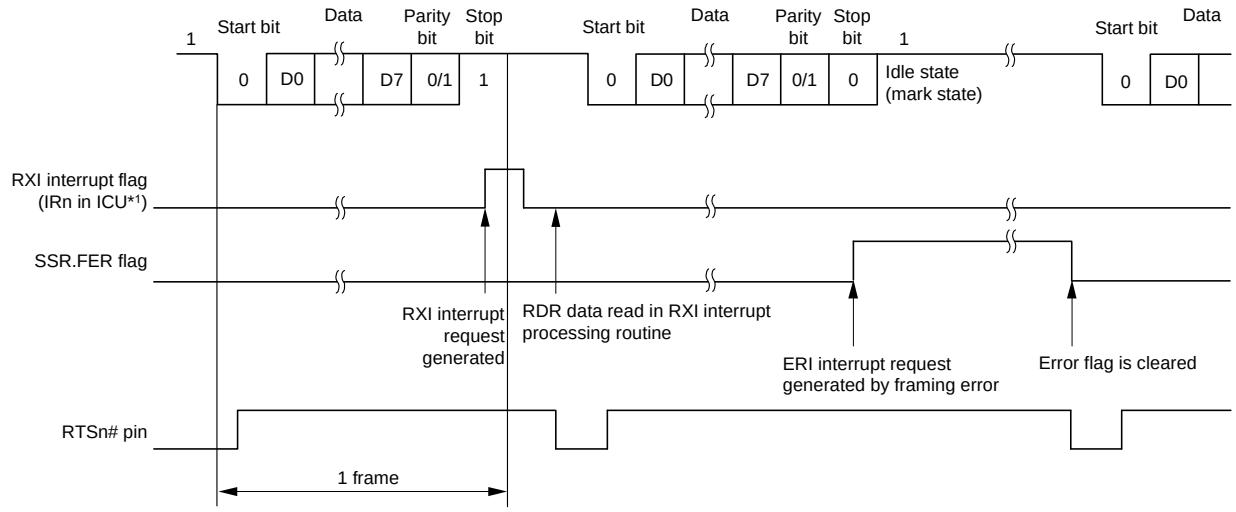
Figure 26.13 and Figure 26.14 show an example of the operation for serial data reception in asynchronous mode. In serial data reception, the SCI operates as described below.

1. When the value of the RE bit in SCR becomes 1, the output signal on the RTSn# pin goes to the low level.
2. When the SCI monitors the communications line and detects a start bit, it performs internal synchronization, stores receive data in RSR, and checks the parity bit and stop bit.
3. If an overrun error occurs, the ORER flag in SSR is set to 1. If the RIE bit in SCR is 1 at this time, an ERI interrupt request is generated. Receive data is not transferred to RDR.
4. If a parity error is detected, the PER bit in SSR is set to 1 and receive data is transferred to RDR. If the RIE bit in SCR is 1 at this time, an ERI interrupt request is generated.
5. If a framing error (when the stop bit is 0) is detected, the FER bit in SSR is set to 1 and receive data is transferred to RDR. If the RIE bit in SCR is 1 at this time, an ERI interrupt request is generated.
6. When reception finishes successfully, receive data is transferred to RDR. If the RIE bit in SCR is 1 at this time, an RXI interrupt request is generated. Continuous reception is enabled by reading the receive data transferred to RDR in this RXI interrupt processing routine before reception of the next receive data is completed. Reading out the received data that have been transferred to RDR causes the RTSn# pin to output the low level.




Note 1. Refer to the Interrupt Controller section for details on the corresponding interrupt vector number

**Figure 26.13 Example of SCI Operation for Serial Reception in Asynchronous Mode (1) (When RTS Function is Not Used) (Example with 8-Bit Data, Parity, 1 Stop Bit)**




Note 1. Refer to the Interrupt Controller section for details on the corresponding interrupt vector number.

**Figure 26.14 Example of SCI Operation for Serial Reception in Asynchronous Mode (2) (When RTS Function is Used) (Example with 8-Bit Data, Parity, 1 Stop Bit)**

Table 26.25 lists the states of the SSR status flags and receive data handling when a receive error is detected.
If a receive error is detected, an ERI interrupt request is generated but an RXI interrupt request is not generated. Data reception cannot be resumed while the receive error flag is 1. Accordingly, clear the ORER, FER, and PER bits to 0 before resuming reception. Moreover, be sure to read the RDR during overrun error processing. When a reception is forcibly terminated by clearing the SCR.RE bit to 0 during operation, read the RDR register because received data which has not yet been read may be left in RDR.
Figure 26.15 and Figure 26.16 show samples of flowcharts for serial data reception.

**Table 26.25 SSR Status Flags and Receive Data Handling**

| SSR Status Flag | | | | |
|---|---|---|---|---|
| **ORER** | **FER** | **PER** | **Receive Data** | **Receive Error Type** |
| 1 | 0 | 0 | Lost | Overrun error |
| 0 | 1 | 0 | Transferred to RDR | Framing error |
| 0 | 0 | 1 | Transferred to RDR | Parity error |
| 1 | 1 | 0 | Lost | Overrun error + framing error |
| 1 | 0 | 1 | Lost | Overrun error + parity error |
| 0 | 1 | 1 | Transferred to RDR | Framing error + parity error |
| 1 | 1 | 1 | Lost | Overrun error + framing error + parity error |

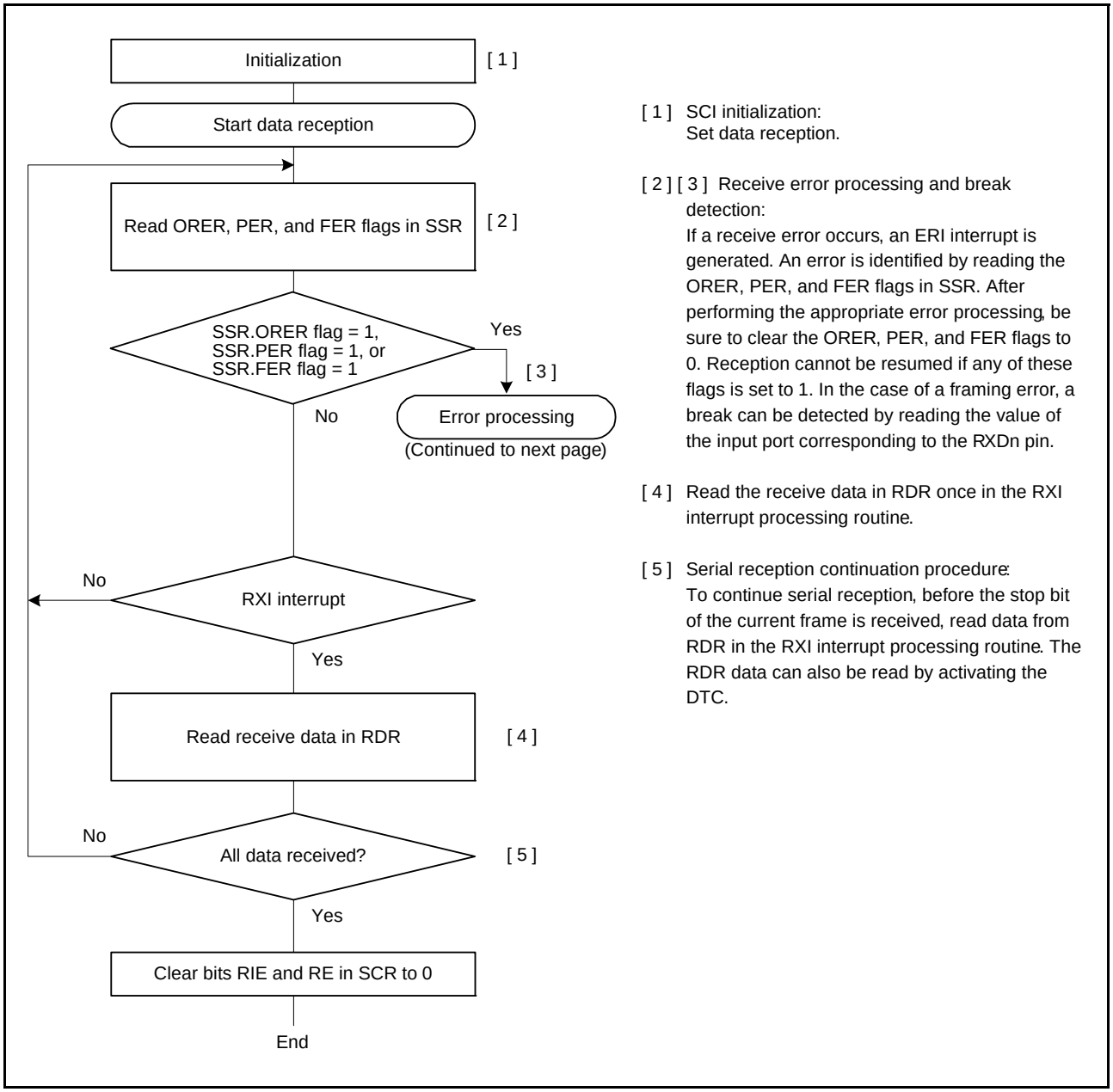



**Figure 26.15 Example Flowchart of Serial Reception in Asynchronous Mode (1)**

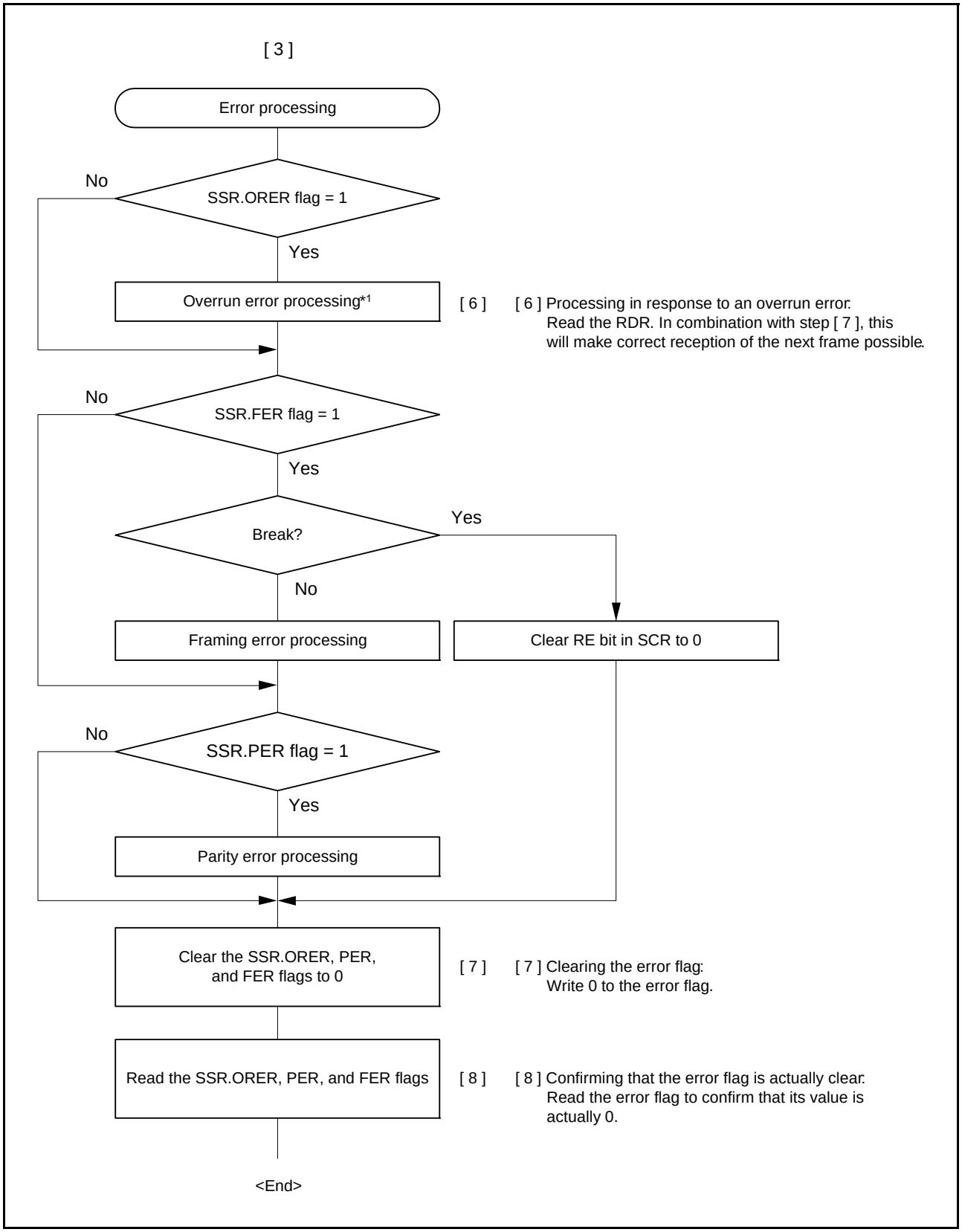



**Figure 26.16 Example Flowchart of Serial Reception in Asynchronous Mode (2)**

## 26.4 Multi-Processor Communications Function

Using the multi-processor communication functions enables to transmit and receive data by sharing a communication line between multiple processors by using asynchronous serial communication in which the multi-processor bit is added. In multi-processor communication, a unique ID code is allocated to each receiving station. Serial communication cycles consist of an ID transmission cycle to specify the receiving station and a data transmission cycle to transmit data to the specified receiving station. The multi-processor bit is used to distinguish between the ID transmission cycle and the data transmission cycle. When the multi-processor bit is set to 1, it indicates the ID transmission cycle and when the multi-processor bit is set to 0, it indicates the data transmission cycle. Figure 26.17 shows an example of communication between processors by using a multi-processor format. First, a transmitting station transmits communication data in which the multi-processor bit set to 1 is added to the ID code of the receiving station. Next, the transmitting station transmits the communication data in which the multi-processor bit set to 0 is added to the transmit data. Upon receiving the communication data in which the multi-processor bit is set to 1, the receiving station compares the received ID with the ID of the receiving station itself and if the two match, receives the communication data that is subsequently transmitted. If the received ID does not match with the ID of the receiving station, the receiving station skips the communication data until again receiving the communication data in which the multi-processor bit is set to 1.

For supporting this function, the SCI provides the MPIE bit in SCR. When the MPIE bit is set to 1, transfer of receive data from the RSR to the RDR, detection of a reception error, and setting the respective status flags ORER and FER in SSR are disabled until reception of data in which the multi-processor bit is set to 1. Upon receiving a reception character in which the multi-processor bit is set to 1, the MPBT bit in SSR is set to 1 and the MPIE bit in SCR is automatically cleared, thus returning to a normal reception operation. During this time, an RXI interrupt is generated if the RIE bit in SCR is set.

When the multi-processor format is specified, specification of the parity bit is disabled. Apart from this, there is no difference from the operation in the normal asynchronous mode. A clock which is used for the multi-processor communication is also the same as the clock used in the normal asynchronous mode.

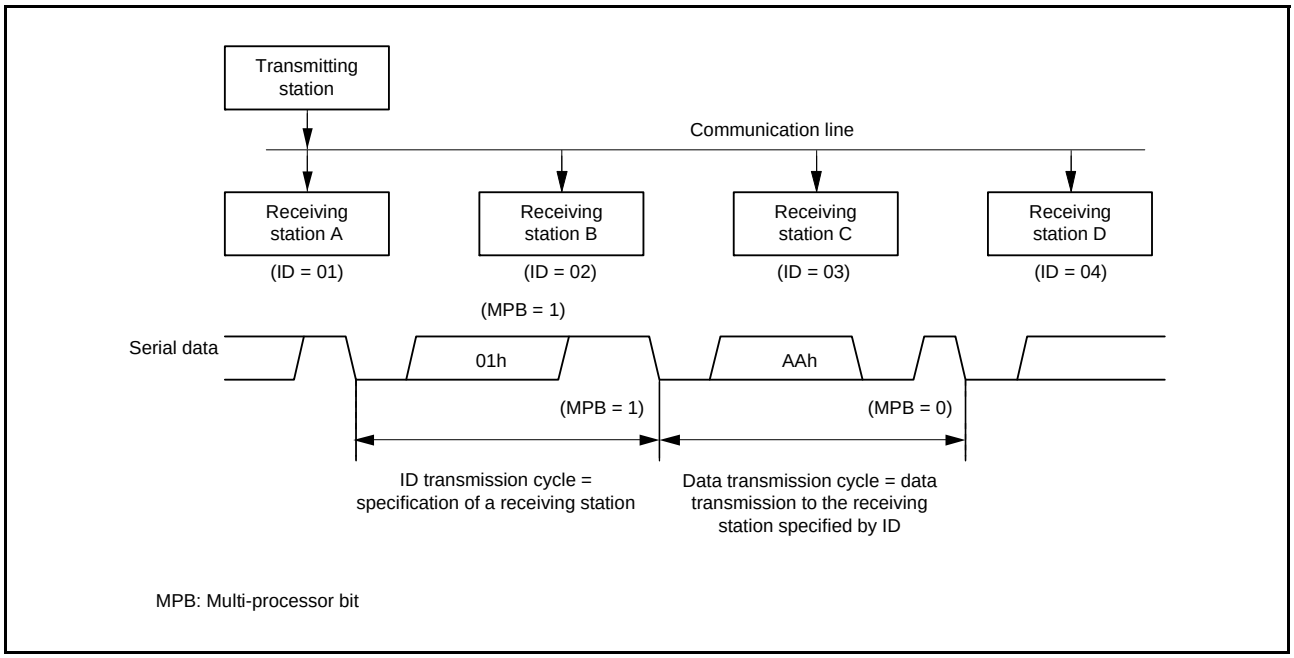



**Figure 26.17 An Example of Communication using the Multi-Processor Format (Example of Transmission of Data AAh to Receiving Station A)**

### 26.4.1 Multi-Processor Serial Data Transmission

Figure 26.18 is a sample flowchart of multi-processor data transmission. In the ID transmission cycle, the ID should be transmitted with the MPBT bit in SSR set to 1. In the data transmission cycle, the data should be transmitted with the MPBT bit set to 0. The other operations are the same as the operations in asynchronous mode.

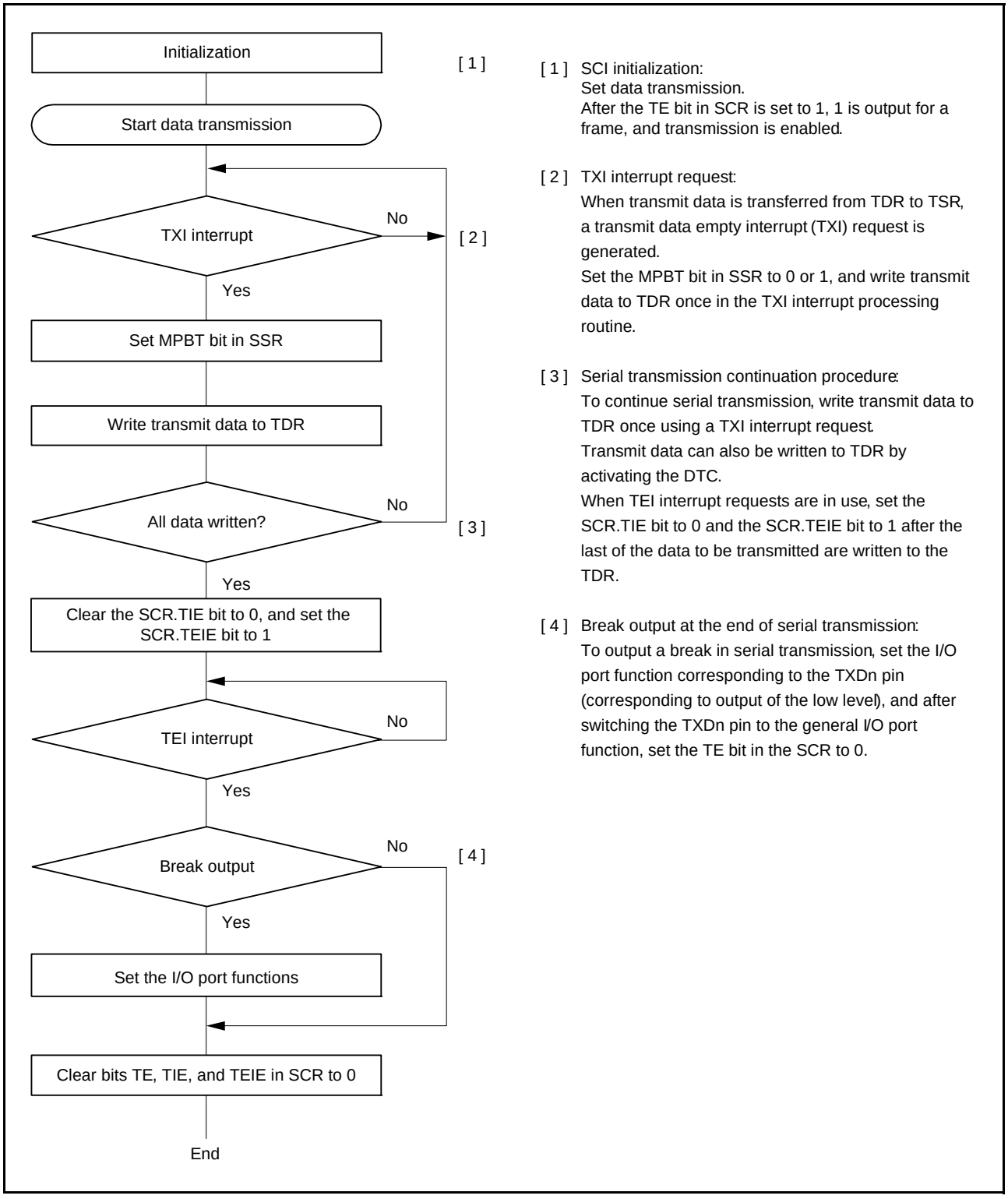



**Figure 26.18 Example of Multi-Processor Serial Transmission Flowchart**

## 26.4.2 Multi-Processor Serial Data Reception

Figure 26.20 and Figure 26.21 are sample flowcharts of multi-processor data reception. When the MPIE bit in SCR is set to 1, reading the communication data is skipped until reception of the communication data in which the multi-processor bit is set to 1. When the communication data in which the multi-processor bit is set to 1 is received, the received data is transferred to RDR. During this time, the RXI interrupt request is generated. The other operations are the same as the operations in asynchronous mode.
Figure 26.19 is the example of operation for reception.

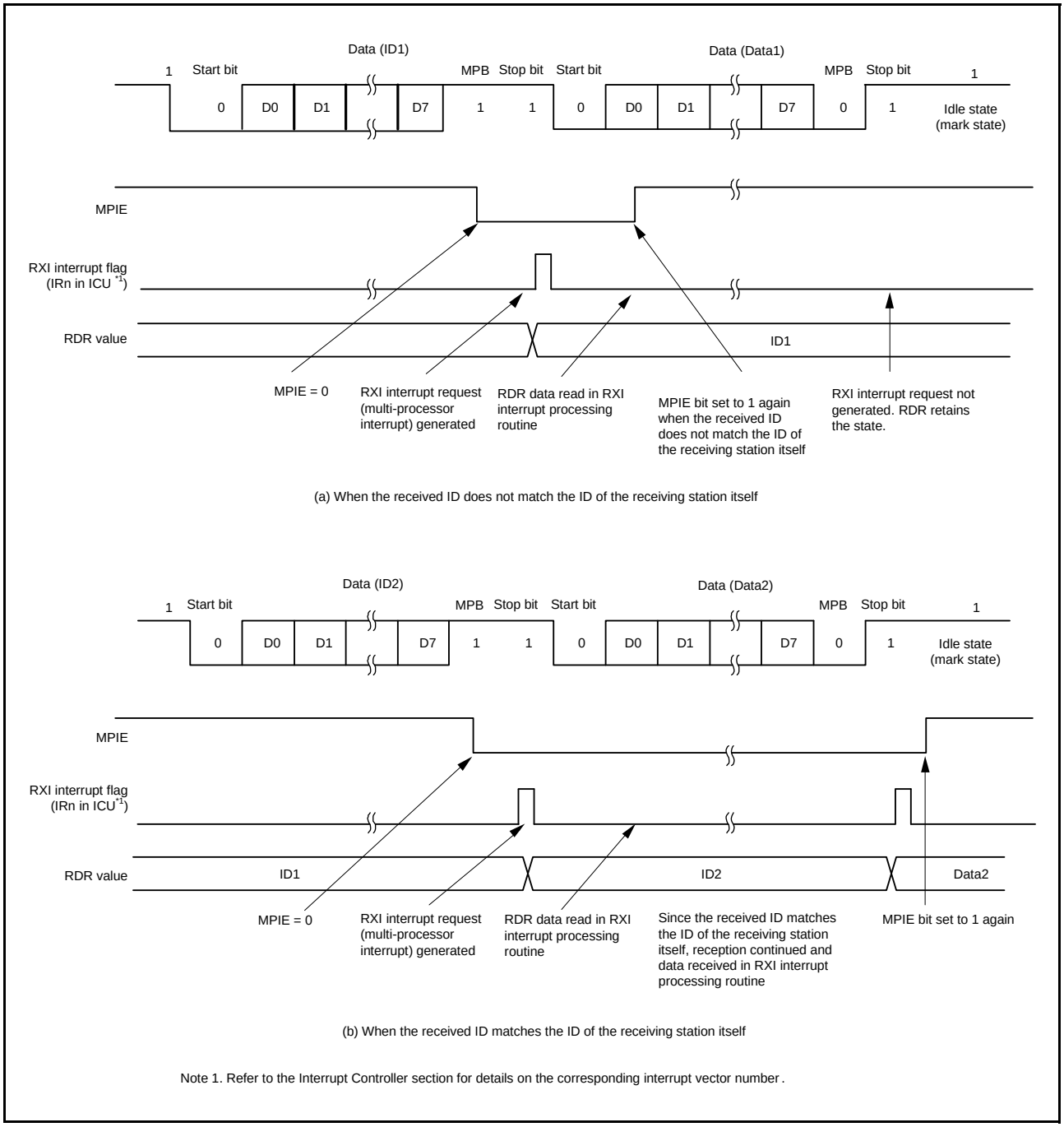



**Figure 26.19 Example of SCI Reception (8-Bit Data/Multi-Processor Bit/1 Stop Bit)**

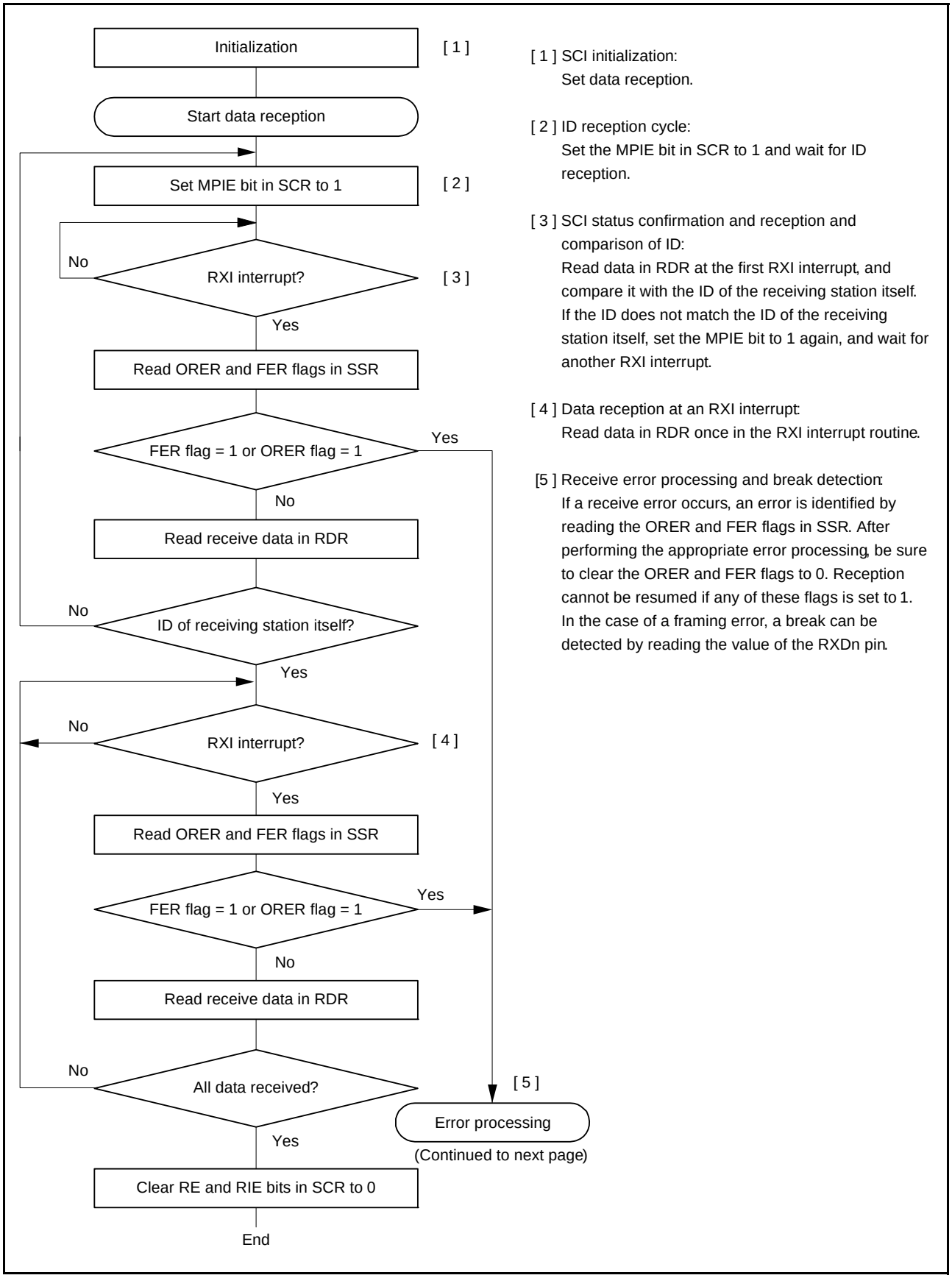



**Figure 26.20 Example of Multi-Processor Serial Reception Flowchart (1)**

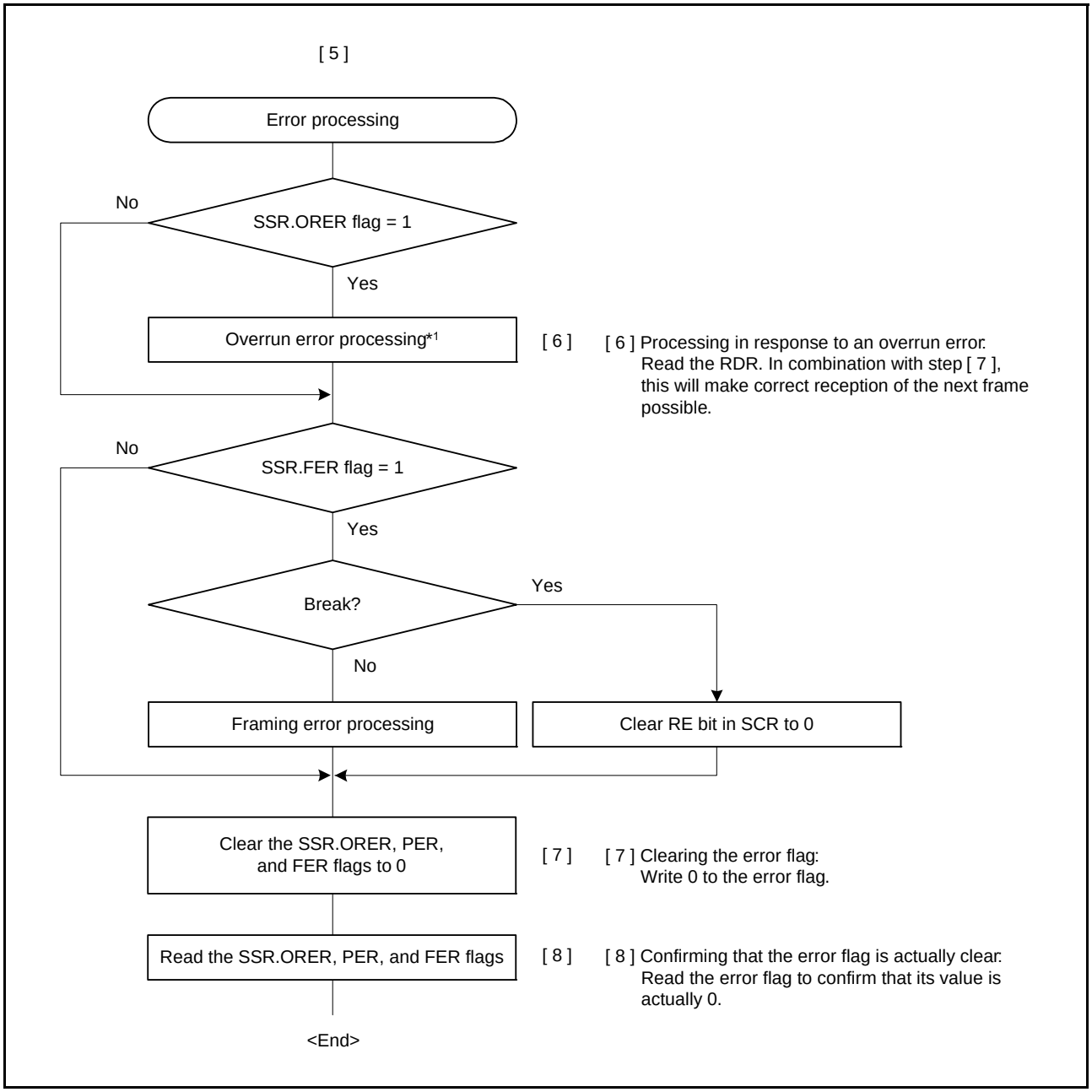



**Figure 26.21 Example of Multi-Processor Serial Reception Flowchart (2)**

## 26.5 Operation in Clock Synchronous Mode

Figure 26.22 shows the data format for clock synchronous serial data communications.
In clock synchronous mode, data is transmitted or received in synchronization with clock pulses. One character in transfer data consists of 8-bit data. In clock synchronous mode, no parity bit can be added.
In data transmission, the SCI outputs data from one falling edge of the synchronization clock to the next. In data reception, the SCI receives data in synchronization with the rising edge of the synchronization clock. After 8-bit data is output, the transmission line holds the last bit output state.
Inside the SCI, the transmitter and receiver are independent units, enabling full-duplex communications by use of a common clock. Both the transmitter and the receiver also have a double-buffered structure, so that the next transmit data can be written during transmission or the previous receive data can be read during reception, enabling continuous data transfer.

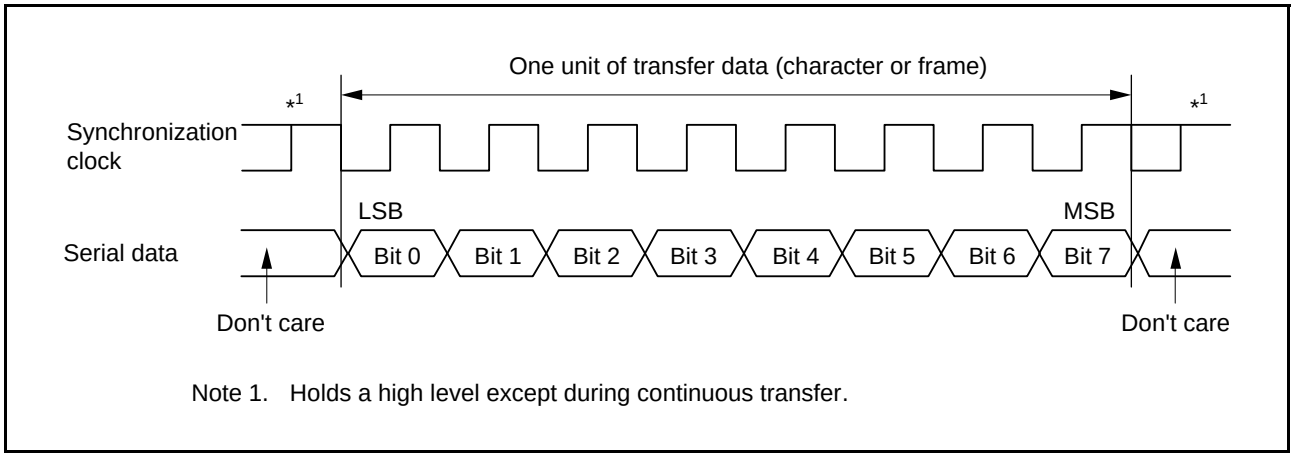



**Figure 26.22 Data Format in Clock Synchronous Serial Communications (LSB First)**

### 26.5.1 Clock

Either an internal clock generated by the on-chip baud rate generator or an external synchronization clock input at the SCKn pin can be selected, according to the setting of the CKE[1:0] bits in SCR.
When the SCI is operated on an internal clock, the synchronization clock is output from the SCKn pin. Eight synchronization clock pulses are output in the transfer of one character, and when no transfer is performed the clock is held high. However, when only data reception is performed while the CTS function is disabled, the synchronization clock output is started at the same time when the SCR.RE bit set to 1. The synchronization clock is stopped at the high level when an overrun error occurs or the SCR.RE bit is cleared to 0.
When only data reception is performed and the CTS function is enabled, the clock output is not started even when the SCR.RE bit set to 1 if the CTSn# pin input is high when the SCR.RE bit is 0. The synchronization clock output is started when the SCR.RE bit is set to 1 and the CTSn# pin input is low. After that, if the CTSn# pin input is high on completion of the frame reception, the synchronization clock output is stopped at the high level. If the CTSn# pin input continues to be low, the synchronization clock is stopped at the high level when an overrun error occurs or the SCR.RE bit is cleared to 0.

### 26.5.2 CTS and RTS Functions

In the CTS function, CTSn# pin input is used to control reception/transmission start when the clock source is the internal clock. Setting the SPMR.CTSE bit to 1 enables the CTS function.
When the CTS function is enabled, placing the low level on the CTSn# pin causes reception/transmission to start.
In the RTS function, RTSn# pin output is used to request reception/transmission start when the clock source is an external synchronizing clock. A low level is output when serial communications become possible. Conditions for output of the low and high level are shown below.

[Conditions for low-level output]
Satisfaction of all conditions listed below

- The value of the RE or TE bit in the SCR is 1
- neither transmission nor reception is in progress
- there are no received data yet to be read (when the SCR.RE bit is 1)
- transmit data has been written (when the SCR.TE bit is 1)
- ORER flag in SSR is 0

[Condition for high-level output]
The conditions for low-level output have not been satisfied.

### 26.5.3 SCI Initialization (Clock Synchronous Mode)

Before transmitting and receiving data, start by writing the initial value 00h to the SCR and then continue through the procedure for SCI given in Figure 26.23. Whenever the operating mode or transfer format is changed, the SCR must be initialized before the change is made.
Note that clearing the SCR.RE bit to 0 initializes neither the ORER, FER, and PER flags in SSR nor RDR.
Moreover, note that switching the value of the SCR.TE bit from 1 to 0 or 0 to 1 while the SCR.TIE bit is 1 leads to the generation of a TXI interrupt request.

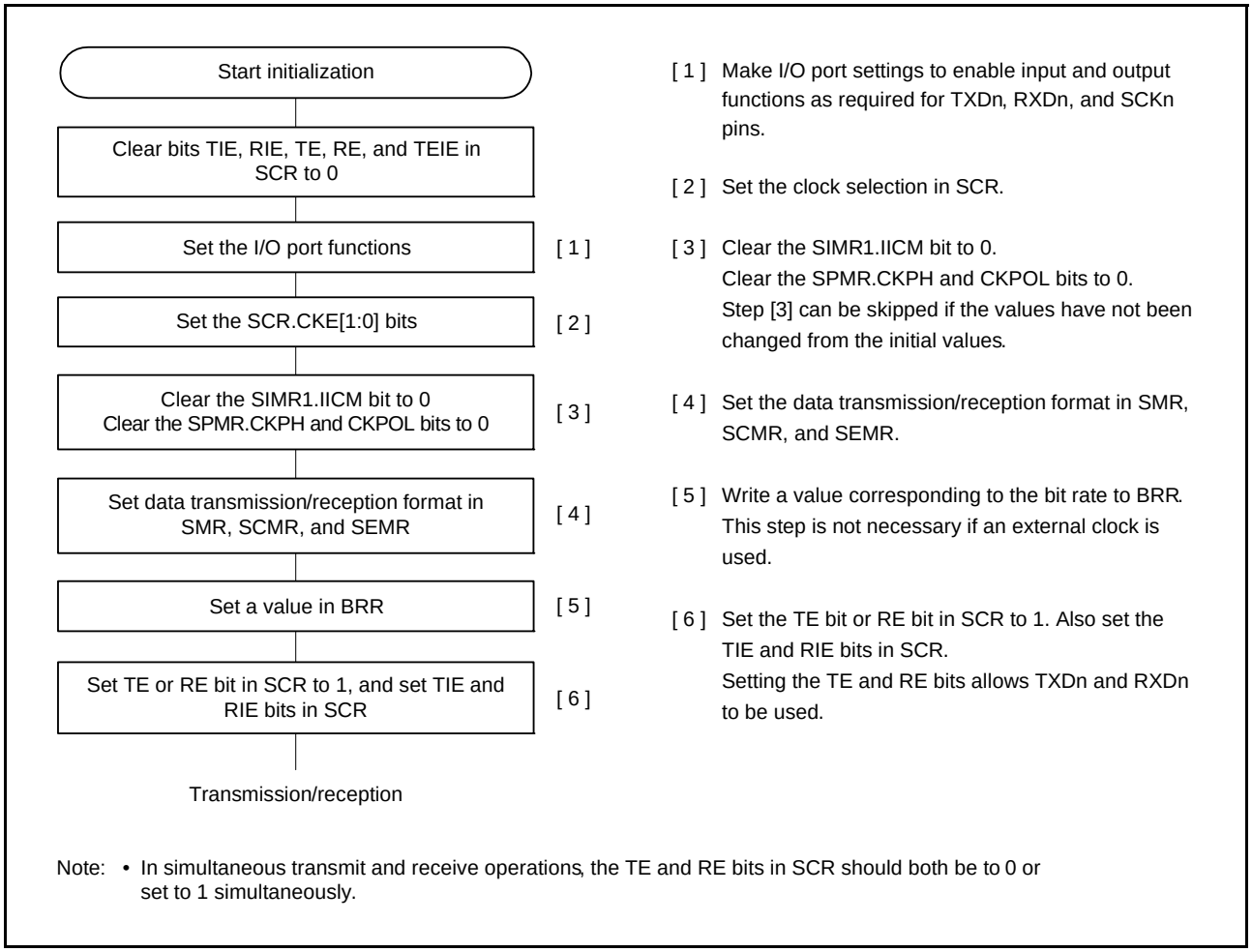


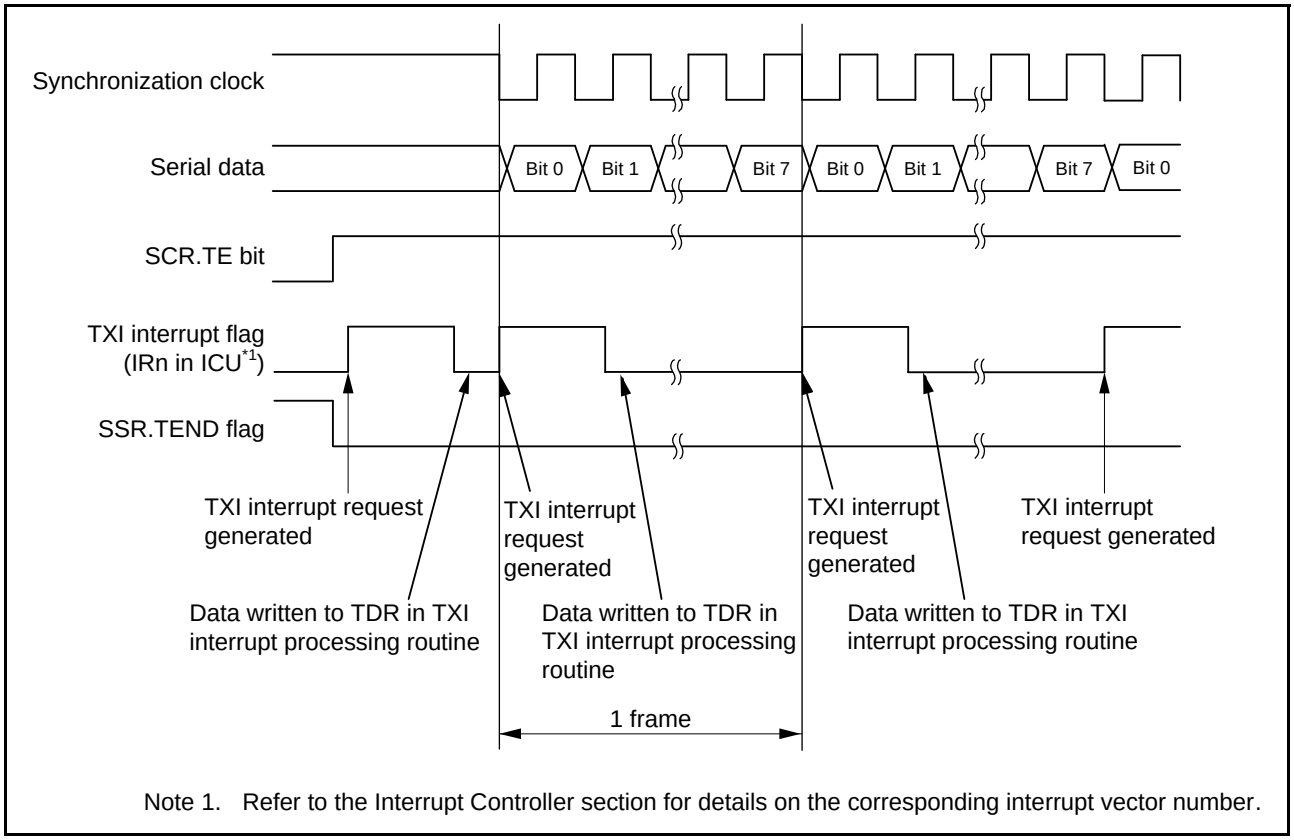

**Figure 26.23 Example of SCI Initialization Flowchart (Clock Synchronous Mode)**

### 26.5.4 Serial Data Transmission (Clock Synchronous Mode)

Figure 26.23, Figure 26.24, and Figure 26.25 show an example of the operation for serial transmission in clock synchronous mode.
In serial data transmission, the SCI operates as described below.

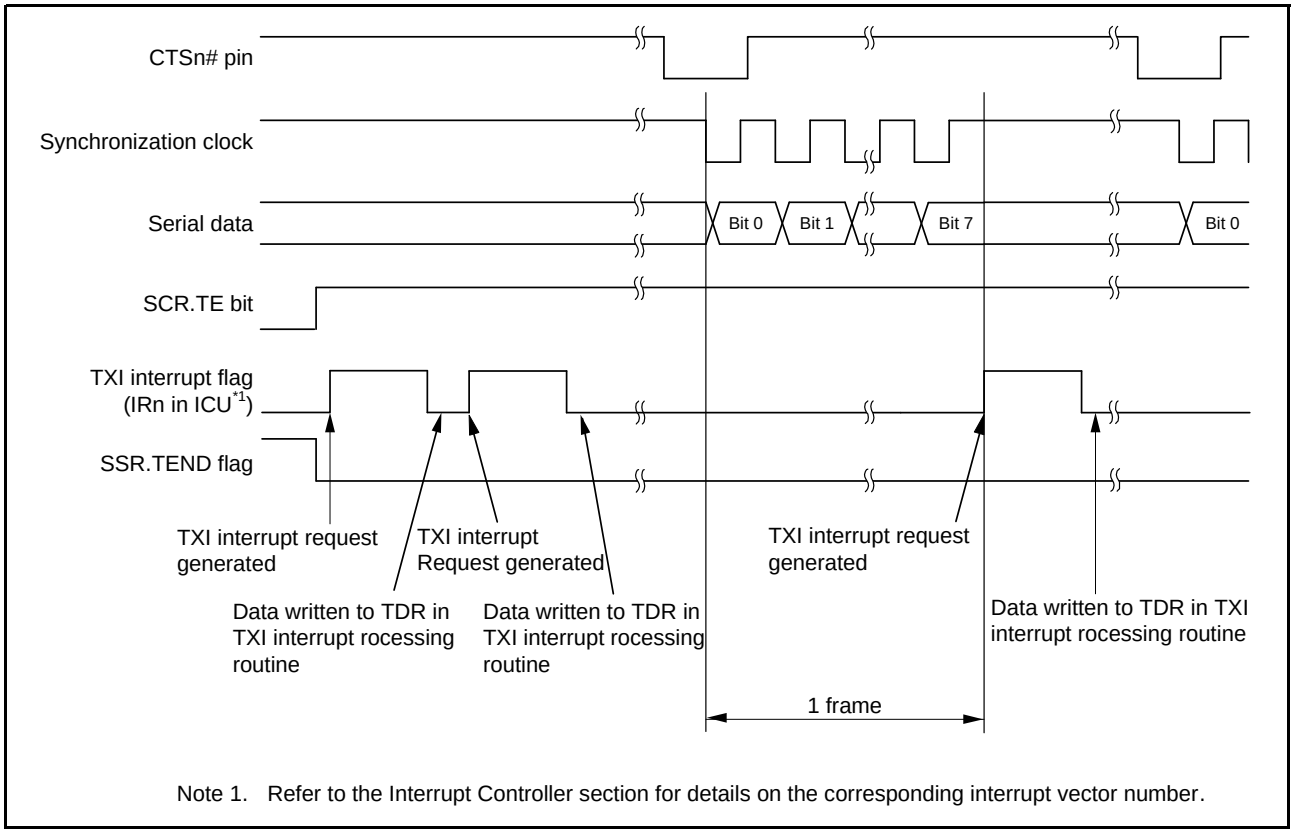
1. The SCI transfers data from TDR to TSR when data is written to TDR in the TXI interrupt processing routine. The TXI interrupt request at the beginning of transmission is generated when the TE bit in SCR is set to 1 after the TIE bit in SCR is set to 1 or when these two bits are set to 1 simultaneously by a single instruction.
2. After transferring data from TDR to TSR, the SCI starts transmission. When the SCR.TIE bit is set to 1 at this time, a TXI interrupt request is generated. Continuous transmission is enabled by writing the next transmit data to TDR in this TXI interrupt processing routine before transmission of the current transmit data has finished. When TEI interrupt requests are in use, set the SCR.TIE bit to 0 (a TXI interrupt request is disabled) and the SCR.TEIE bit to 1 (a TEI interrupt request is enabled) after the last of the data to be transmitted are written to the TDR from the processing routine for TXI requests.
3. 8-bit data is sent from the TXDn pin in synchronization with the output clock when clock output mode has been specified and in synchronization with the input clock when use of an external clock has been specified. Output of the clock signal is suspended until the input CTS signal is at the low level while the CTSE bit in SPMR is 1 (CTS function is enabled).
4. The SCI checks for updating of (writing to) the TDR at the time of the last bit output.
5. When TDR is updated, the next transmit data is transferred from TDR to TSR, and serial transmission of the next frame is started.
6. If TDR is not updated, set the SSR.TEND flag to 1 and the TXDn pin retains the output state of the last bit. If the TEIE bit in SCR is 1 at this time, a TEI interrupt request is generated. The SCKn pin is held high.

Figure 26.27 shows a sample flowchart of serial data transmission.
Transmission will not start while a receive error flag (ORER, FER, or PER in SSR) is set to 1. Be sure to clear the receive error flags to 0 before starting transmission. Note that clearing the RE bit in SCR to 0 does not clear the receive error flags.




Note 1. Refer to the Interrupt Controller section for details on the corresponding interrupt vector number.

**Figure 26.24 Example of Operation for Serial Transmission in Clock Synchronous Mode (1) (CTS Function is not Used, at the Beginning of Transmission)**




Note 1. Refer to the Interrupt Controller section for details on the corresponding interrupt vector number.

**Figure 26.25 Example of Operation for Serial Transmission in Clock Synchronous Mode (2) (CTS Function is Used, at the Beginning of Transmission)**

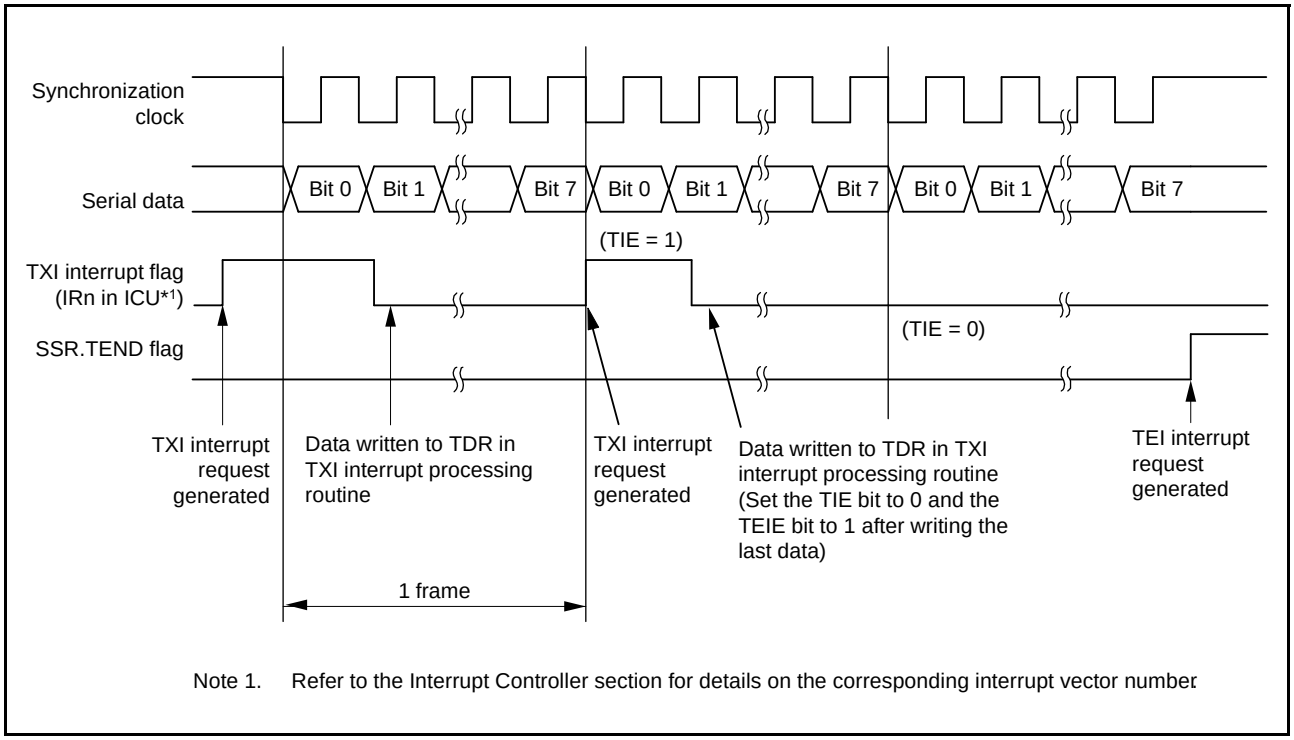
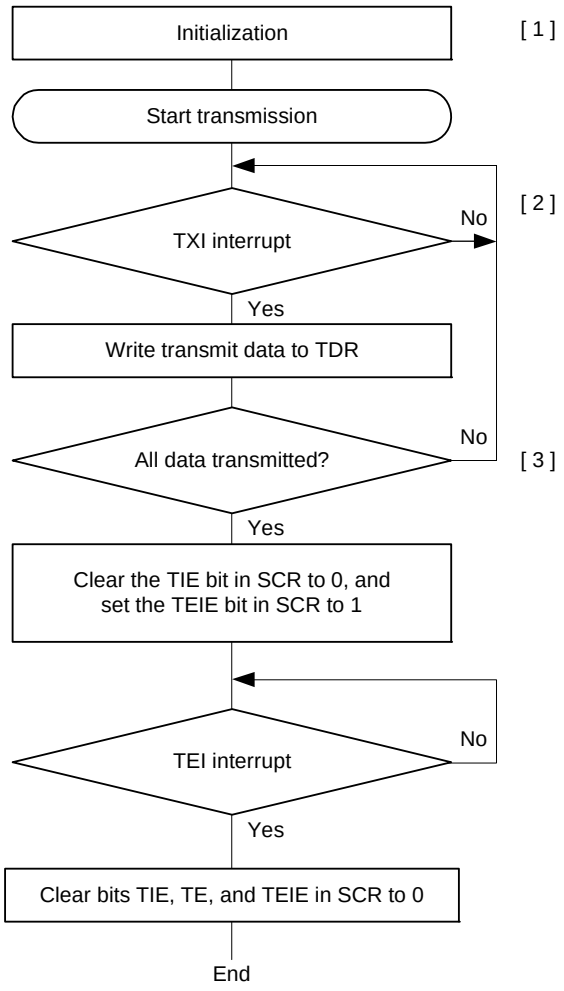



**Figure 26.26 Example of Operation for Serial Transmission in Clock Synchronous Mode (3) (from the Middle of Transmission until Transmission Completion)**




[ 1 ] SCI initialization:
Set data transmission.

[ 2 ] Writing transmit data write to TDR by a TXI interrupt request:
When transmit data is transferred from TDR to TSR, a transmit data empty interrupt (TXI) request is generated.
Transmit data is written to TDR once from the processing routine for TXI requests.

[ 3 ] Serial transmission continuation procedure:
To continue serial transmission, write transmit data to TDR upon accepting a transmit data empty interrupt (TXI). Transmit data can also be written to TDR by activating the DTC by the TXI interrupt request.
When TEI interrupt requests are in use, set the SCR.TIE bit to 0 and the SCR.TEIE bit to 1 after the last of the data to be transmitted are written to the TDR.

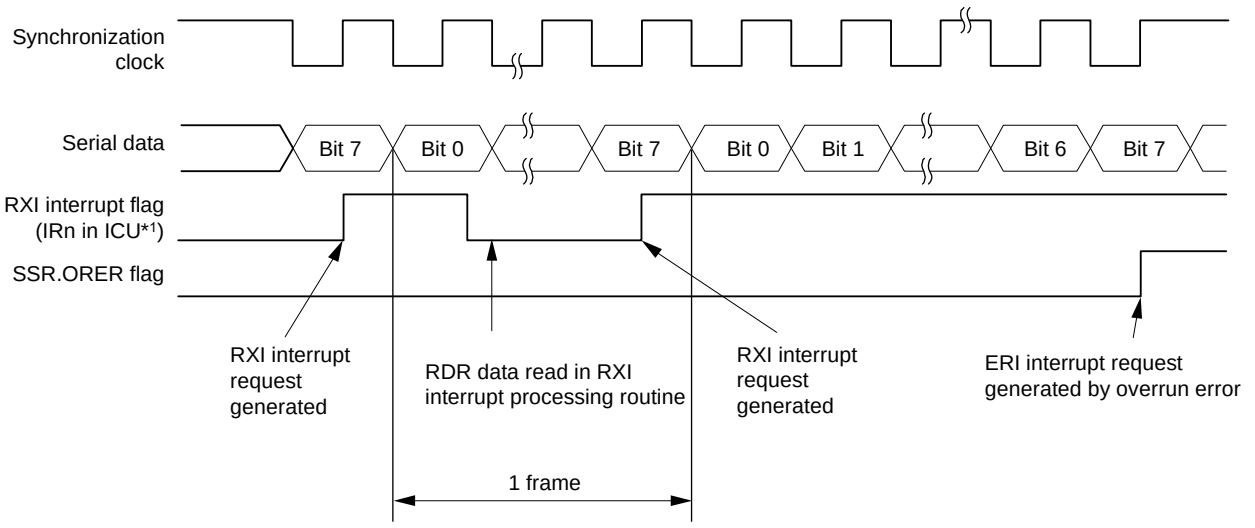
Note: • When the external clock is in use (the value of the SCR.CKE[1:0] bits is 10b or 11b), the rising edge on the SCK pin for the last bit sets the SSR.TEND flag to 1. Clearing the SCR.TE bit to 0 immediately after this may lead to insufficient received-data hold time on the receiver side.

**Figure 26.27 Example Flowchart of Serial Transmission in Clock Synchronous Mode**

### 26.5.5 Serial Data Reception (Clock Synchronous Mode)

Figure 26.28 and Figure 26.29 shows an example of SCI operation for serial reception in clock synchronous mode. In serial data reception, the SCI operates as described below.

1. The value of the RE bit in SCR becoming 1 places the signal output on the RTSn# pin at the low level (when the RTS function is in use).
2. The SCI performs internal initialization and starts receiving data in synchronization with a synchronization clock input or output, and stores the receive data in RSR.
3. If an overrun error occurs, the ORER bit in SSR is set to 1. If the RIE bit in SCR is 1 at this time, an ERI interrupt request is generated. Receive data is not transferred to RDR.
4. When reception finishes successfully, receive data is transferred to RDR. If the RIE bit in SCR is 1 at this time, an RXI interrupt request is generated. Continuous reception is enabled by reading the receive data transferred to RDR in this RXI interrupt processing routine before reception of the next receive data is completed. Reading out the received data that have been transferred to RDR causes the RTSn# pin to output the low level (when the RTS function is in use).




Note 1. Refer to the Interrupt Controller section for the corresponding interrupt vector number.

**Figure 26.28 Example of Operation for Serial Reception in Clock Synchronous Mode (1) (When RTS Function is Not Used)**

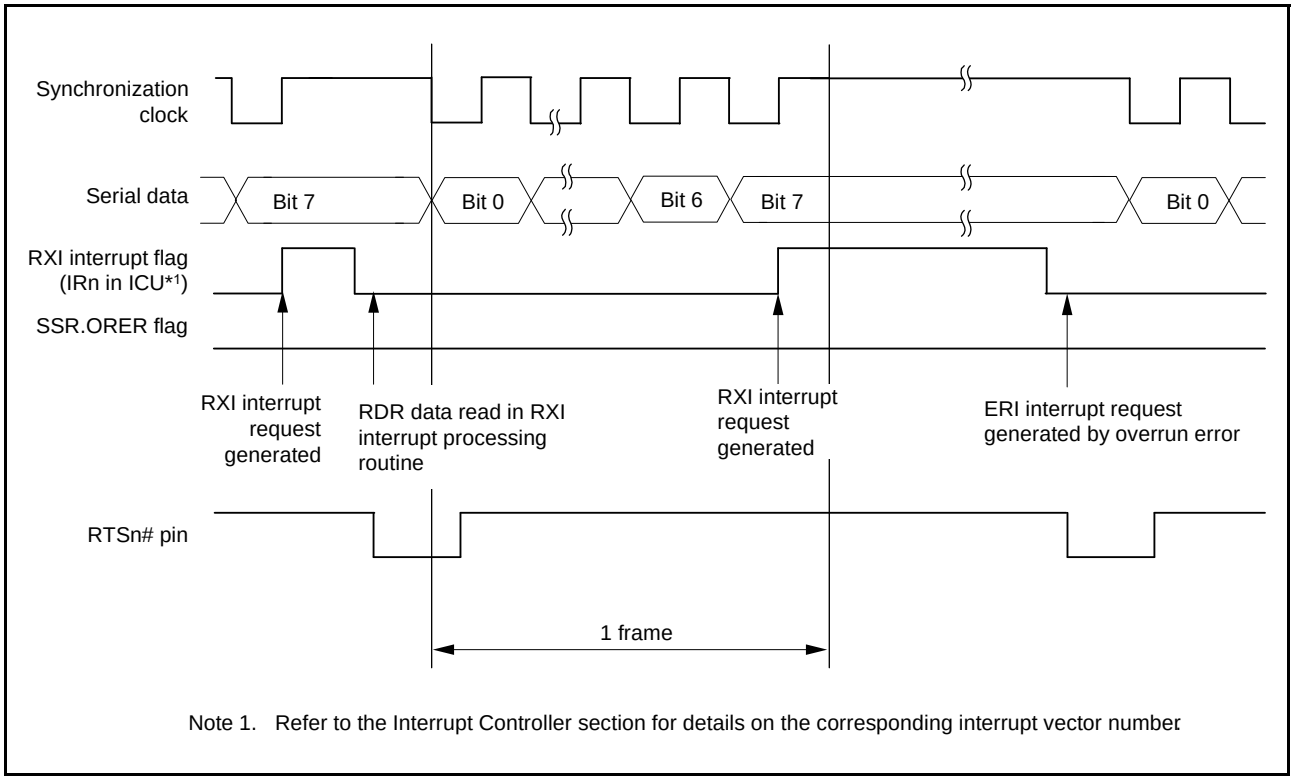
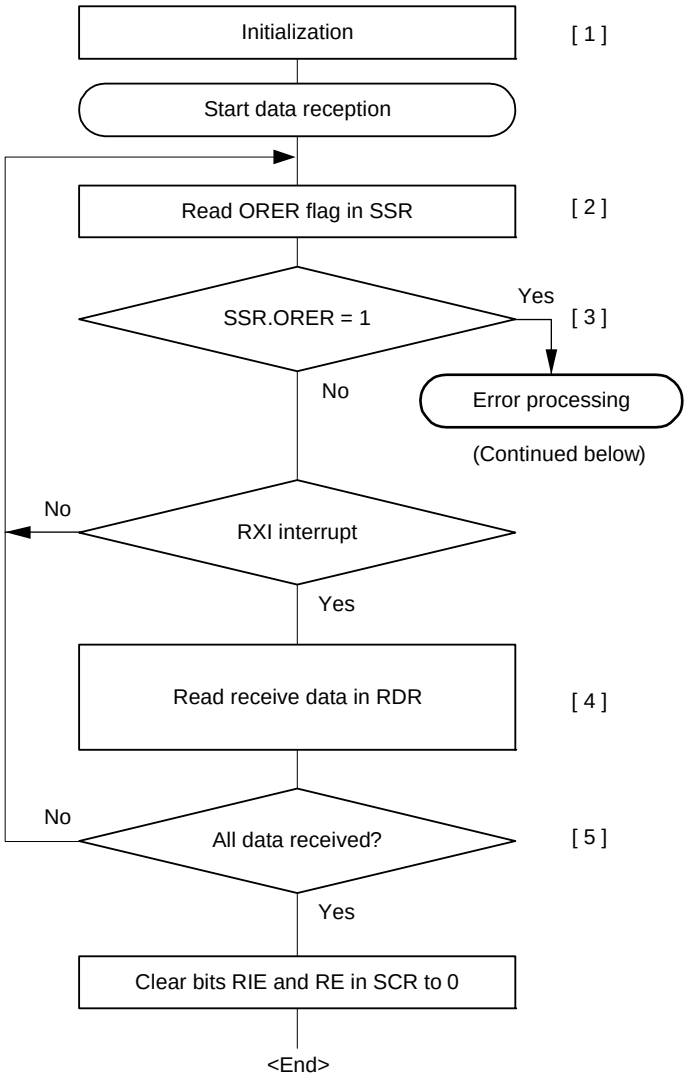
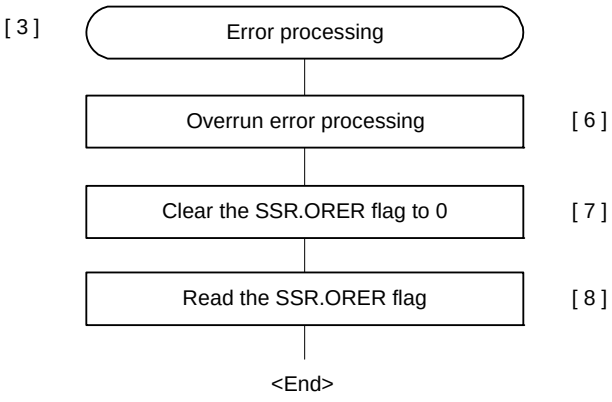



**Figure 26.29 Example of Operation for Serial Reception in Clock Synchronous Mode (2) (When RTS Function is Used)**

Data transfer cannot be resumed while a receive error flag is 1. Accordingly, clear the ORER, FER, and PER bits in SSR to 0 before resuming reception. Moreover, be sure to read the RDR during overrun error processing. When a reception is forcibly terminated by clearing the SCR.RE bit to 0 during operation, read RDR because received data which has not yet been read may be left in RDR.

Figure 26.30 shows a sample flowchart for serial data reception.




[ 1 ] SCI initialization:
Make input port-pin settings for pins to be used as RXDn pins.

[ 2 ] [ 3 ] Receive error processing:
If a receive error occurs, read the ORER flag in SSR, perform the relevant error processing, and then clear the ORER flag to 0. Data reception cannot be resumed while the ORER flag is 1.

[ 4 ] Read the receive data in RDR once in the receive data full interrupt (RXI) request processing routine.

[ 5 ] Serial reception continuation procedure:
To continue serial reception, before the MSB (bit 7) of the current frame is received, finish reading the receive data in RDR. The RDR data can also be read by activating the DTC by an RXI interrupt request.

[ 6 ] Processing in response to an overrun error:
Read the RDR. In combination with step [ 7 ], this will make correct reception of the next frame possible.

[ 7 ] Clearing the error flag:
Write 0 to the error flag.

[ 8 ] Confirming that the error flag is actually clear:
Read the error flag to confirm that its value is actually 0.

**Figure 26.30 Example Flowchart of Serial Reception in Clock Synchronous Mode**

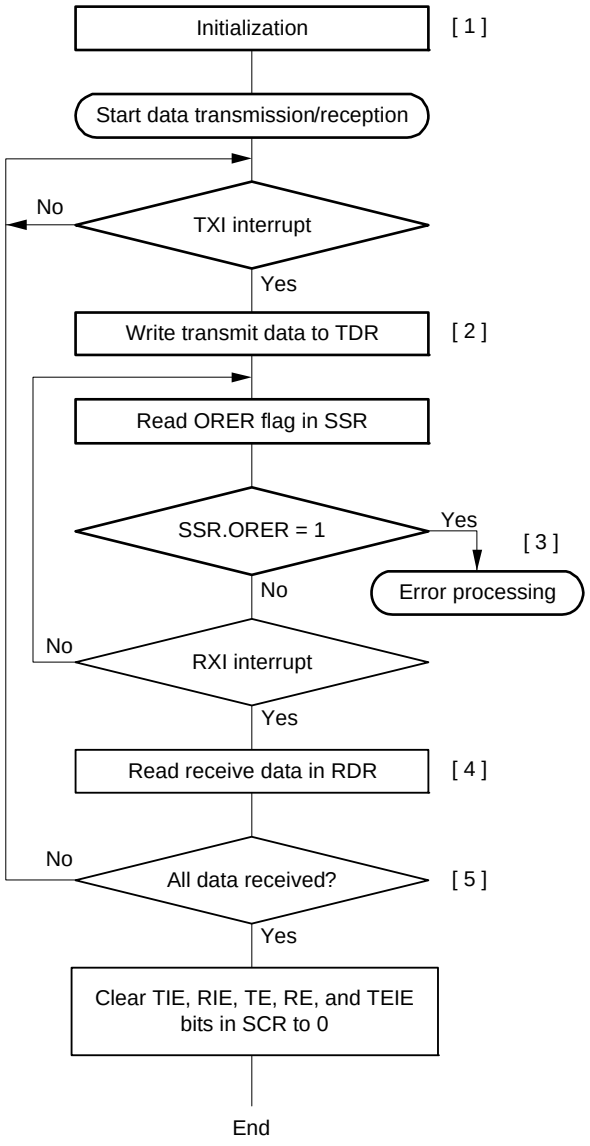
### 26.5.6 Simultaneous Serial Data Transmission and Reception (Clock Synchronous Mode)

Figure 26.31 shows a sample flowchart for simultaneous serial transmit and receive operations in clock synchronous mode.

After initializing the SCI, the following procedure should be used for simultaneous serial data transmit and receive operations.

To switch from transmit mode to simultaneous transmit and receive mode, check that the SCI has finished transmission by reading that the TEND flag in SSR is 1, and then initialize the SCR register. Then set the TIE, RIE, TE, and RE bits in SCR to 1 simultaneously by a single instruction.

To switch from receive mode to simultaneous transmit and receive mode, check that the SCI has finished reception, and then clear the RIE and RE bits to 0. Then check that the receive error flags (ORER, FER, and PER in SSR) are 0, and then set the TIE, RIE, TE, and RE bits in SCR to 1 simultaneously by a single instruction.




[ 1 ] SCI initialization:
The TXDn pin can act as the output pin for transmitted data and the RXDn pin can act as the input pin for received data at the same time.

[ 2 ] Transmit data write:
Write transmit data to TDR once in the TXI interrupt request processing routine.

[ 3 ] Receive error processing:
If a receive error occurs, read the ORER flag in SSR, perform the relevant error processing, and then clear the ORER flag to 0. Data reception cannot be resumed while the ORER flag is 1.

[ 4 ] Reading receive data:
Read the receive data in RDR once in the RXI interrupt request processing routine.

[ 5 ] Serial transmission/reception continuation procedure:
To continue serial transmission and reception, before the MSB (bit 7) of the current frame is received, finish reading the receive data in RDR by the RXI interrupt. Also, before the MSB (bit 7) of the current frame is transmitted, write data to TDR by the TXI interrupt.
Transmit data can also be written to TDR by activating the DTC by a transmit data empty interrupt (TXI) request. Similarly, the RDR data can also be read by activating the DTC by a receive data full interrupt (RXI) request.

Note: • When switching from transmit or receive operation to simultaneous transmit and receive operations, first clear the TIE, RIE, TE, RE, and TEIE bits in SCR to 0, and then set TIE, RIE, TE, and RE bits to 1 simultaneously.

**Figure 26.31 Example Flowchart of Simultaneous Serial Transmission and Reception in Clock Synchronous Mode**

## 26.6 Operation in Smart Card Interface Mode

The SCI supports the smart card (IC card) interface conforming to ISO/IEC 7816-3 (standard for Identification Cards), as an extended function of the SCI.
Smart card interface mode can be selected using the appropriate register.

### 26.6.1 Sample Connection

Figure 26.32 shows a sample connection between a smart card (IC card) and this MCU.
As in the figure, since this MCU communicates with an IC card using a single transmission line, interconnect the TXDn and RXDn pins and pull up the data transmission line to VCC using a resistor.
Setting the TE and RE bits in SCR to 1 with an IC card disconnected enables closed transmission/reception allowing self-diagnosis.
To supply an IC card with the clock pulses generated by the SCI, input the SCKn pin output to the CLK pin of an IC card.
The output port of the this MCU can be used to output a reset signal.

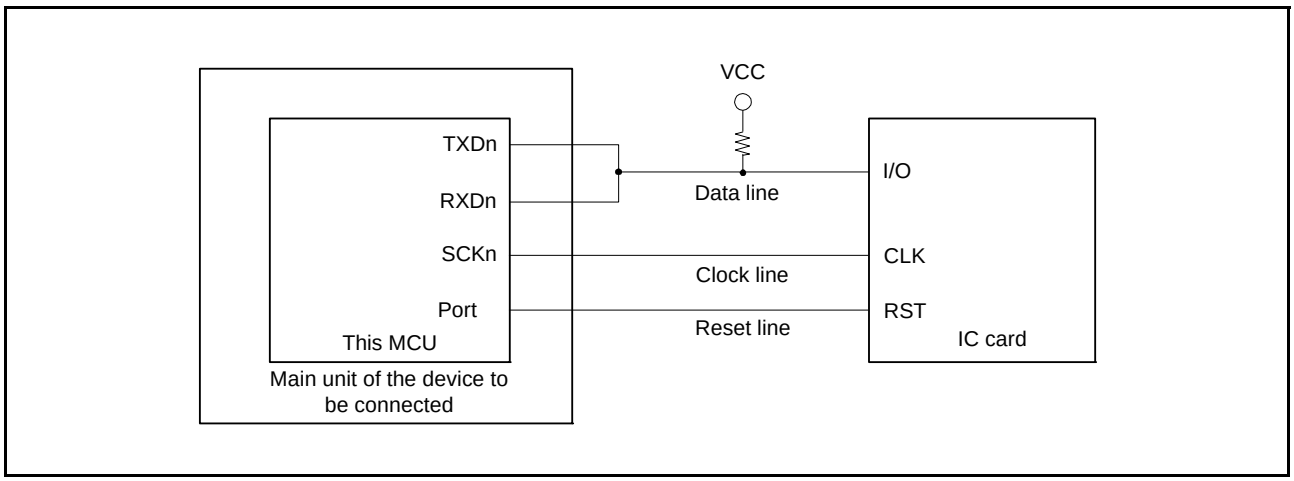



**Figure 26.32 Sample Connection with a Smart Card (IC Card)**

### 26.6.2 Data Format (Except in Block Transfer Mode)

Figure 26.33 shows the data transfer formats in smart card interface mode.

- One frame consists of 8-bit data and a parity bit in asynchronous mode.
- During transmission, at least 2 etu (elementary time unit: time required for transferring 1 bit) is secured as a guard time from the end of the parity bit until the start of the next frame.
- If a parity error is detected during reception, a low-level error signal is output for 1 etu after 10.5 etu has passed from the start bit.
- If an error signal is sampled during transmission, the same data is automatically retransmitted after at least 2 etu.

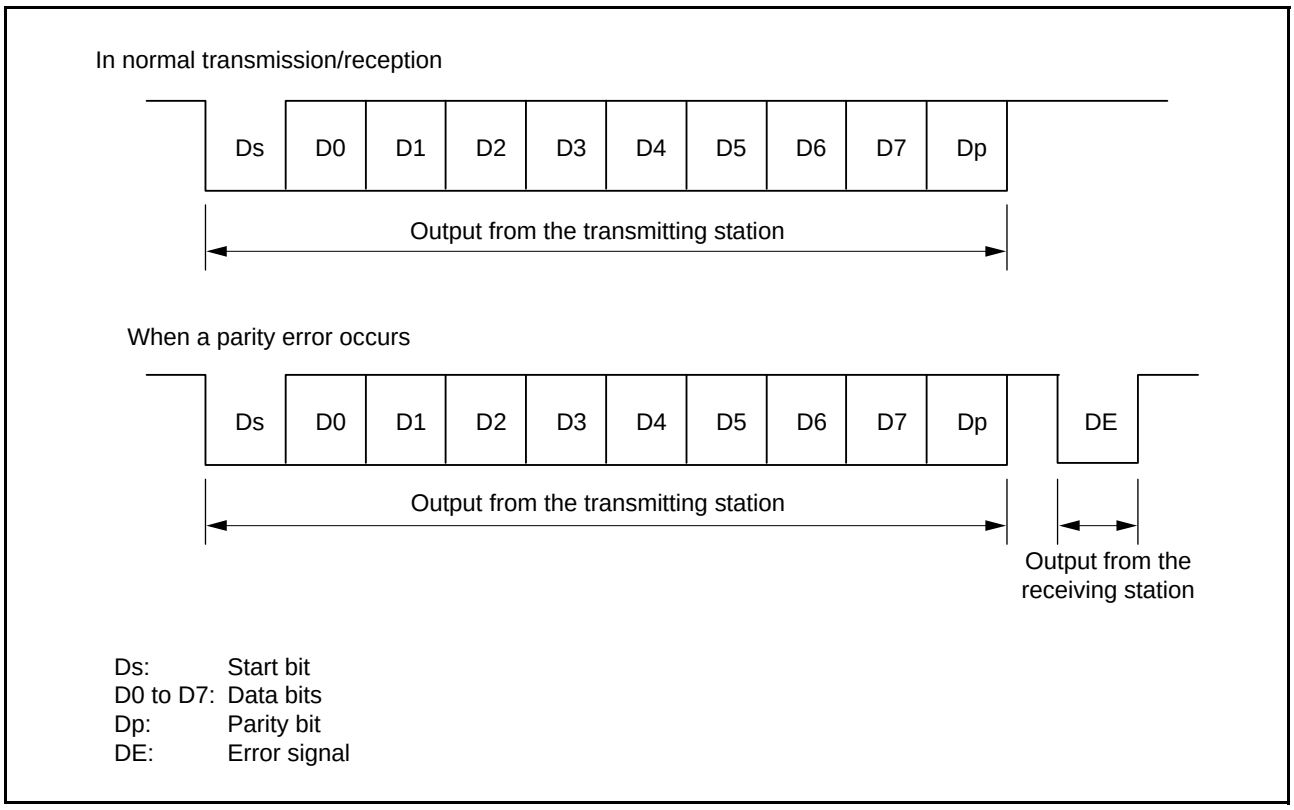



**Figure 26.33 Data Formats in Smart Card Interface Mode**

For communications with IC cards of the direct convention type and inverse convention type, follow the procedure below.

### (1) Direct Convention Type

For the direct convention type, logic levels 1 and 0 correspond to states Z and A, respectively, and data is transferred with LSB first as the start character, as shown in Figure 26.34. Therefore, data in the start character in the figure is 3Bh. When using the direct convention type, write 0 to both the SDIR and SINV bits in SCMR. Write 0 to the PM bit in SMR in order to use even parity, which is prescribed by the smart card standard.

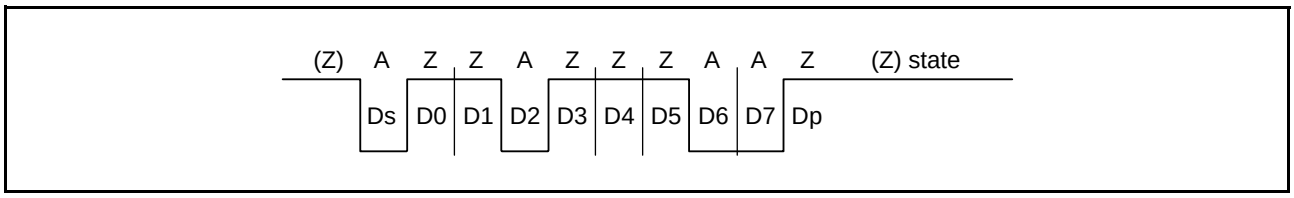



**Figure 26.34 Direct Convention (SDIR in SCMR = 0, SINV in SCMR =0, PM in SMR = 0)**

### (2) Inverse Convention Type

For the inverse convention type, logic levels 1 and 0 correspond to states A and Z, respectively and data is transferred with MSB first as the start character, as shown in Figure 26.35. Therefore, data in the start character in the figure is 3Fh. When using the inverse convention type, write 1 to both the SDIR and SINV bits in SCMR. The parity bit is logic level 0 to produce even parity, which is prescribed by the smart card standard, and corresponds to state Z. Since the SINV bit of the this MCU only inverts data bits D7 to D0, write 1 to the PM bit in SMR to invert the parity bit for both transmission and reception.

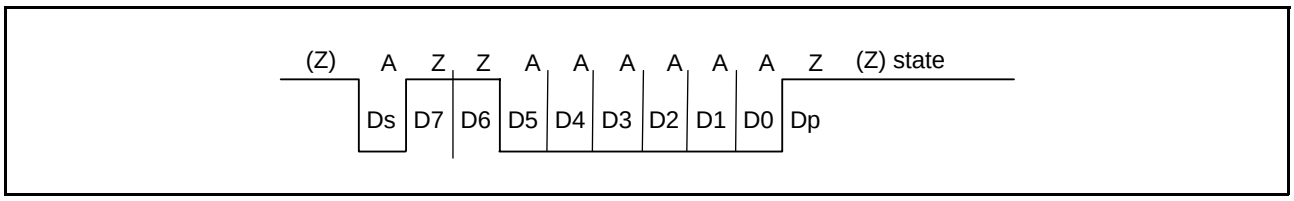



**Figure 26.35 Inverse Convention (SDIR in SCMR = 1, SINV in SCMR =1, PM in SMR = 1)**

## 26.6.3 Block Transfer Mode

Block transfer mode is different from normal smart card interface mode in the following respects.

- Even if a parity error is detected during reception, no error signal is output. Since the PER bit in SSR is set by error detection, clear the PER bit before receiving the parity bit of the next frame.
- During transmission, at least 1 etu is secured as a guard time from the end of the parity bit until the start of the next frame.
- Since the same data is not retransmitted during transmission, the TEND flag in SSR is set 11.5 etu after transmission start.
- In block transfer mode, the ERS flag in SSR indicates the error signal status as in normal smart card interface mode, but the flag is always read as 0 because no error signal is transferred.

### 26.6.4 Receive Data Sampling Timing and Reception Margin

Only the internal clock generated by the on-chip baud rate generator can be used as a transfer clock in smart card interface mode.

In this mode, the SCI can operate on a base clock with a frequency of 32, 64, 372, 256, 93, 128, 186, or 512 times the bit rate according to the settings of the BCP2 bit in SCMR and the BCP[1:0] bits in SMR (the frequency is always 16 times the bit rate in normal asynchronous mode).

For data reception, the falling edge of the start bit is sampled with the base clock to perform internal synchronization. Receive data is sampled on the 16th, 32nd, 186th, 128th, 46th, 64th, 93rd, and 256th rising edges of the base clock so that it can be latched at the middle of each bit as shown in Figure 26.36. The reception margin here is determined by the following formula.

$$M = \left| (0.5 - \frac{1}{2N}) - (L - 0.5)\ F - \frac{D - 0.5}{N}\ (1 + F) \right| \times 100\ [\%]$$

M: Reception margin (%)
N: Ratio of bit rate to clock (N = 32, 64, 372, 256)
D: Duty cycle of clock (D = 0 to 1.0)
L: Frame length (L = 10)
F: Absolute value of clock frequency deviation

Assuming values of F = 0, D = 0.5, and N = 372 in the above formula, the reception margin is determined by the formula below.

$$M = \{0.5 - 1/(2 \times 372)\} \times 100\ [\%] = 49.866\%$$

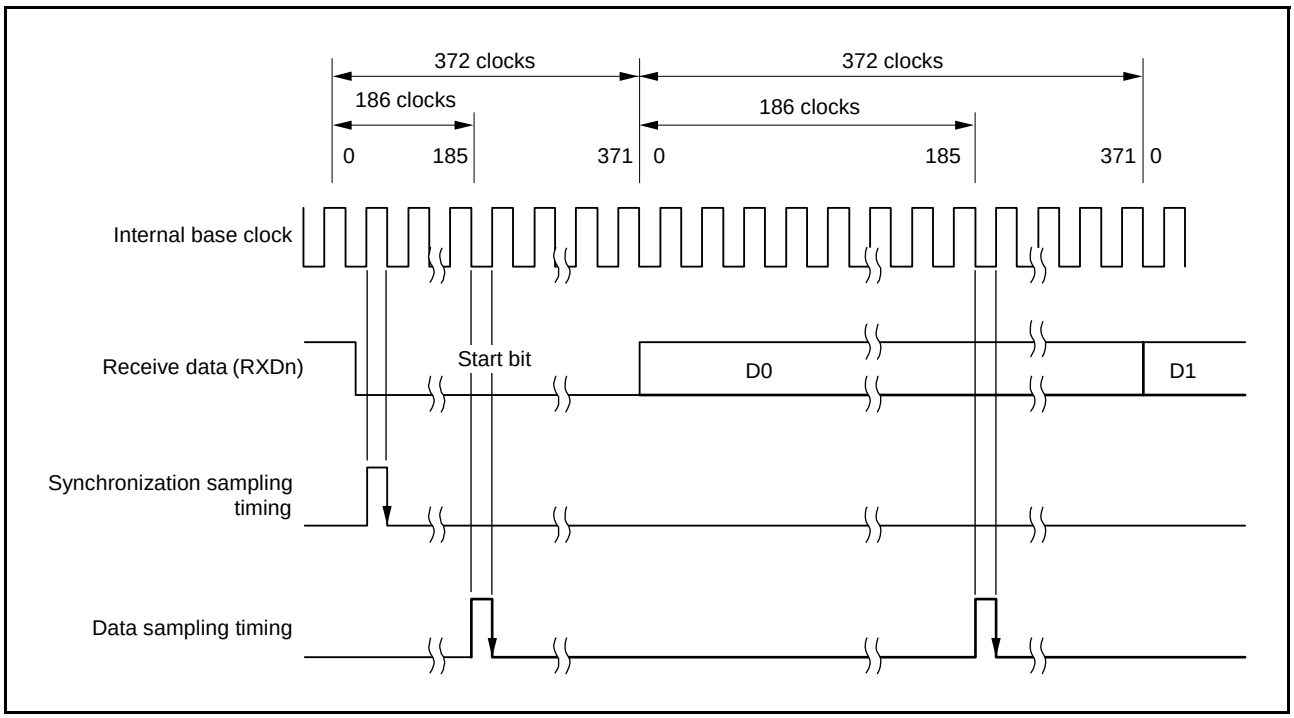



**Figure 26.36 Receive Data Sampling Timing in Smart Card Interface Mode (When Clock Frequency is 372 Times the Bit Rate)**

### 26.6.5 Initialization of the SCI (Smart Card Interface Mode)

Before transmitting and receiving data, write the initial value 00h in the SCR register and initialize the SCI following the example of flowchart shown in Figure 26.37.
Be sure to set the initial value in the SCR register before switching from transmission mode to reception mode and vice versa. Even if the RE bit is cleared to 0, the RDR register is not initialized.

To change reception mode to transmission mode, first check that reception has completed, and then initialize the SCI. At the end of initialization, set TE = 1 and RE = 0. Reception completion can be verified by reading the RXI request, ORER, or PER flag in SSR.
To change transmission mode to reception mode, first check that transmission has completed, and then initialize the SCI. At the end of initialization, set TE = 0 and RE = 1. Transmission completion can be verified by reading the TEND flag in SSR.

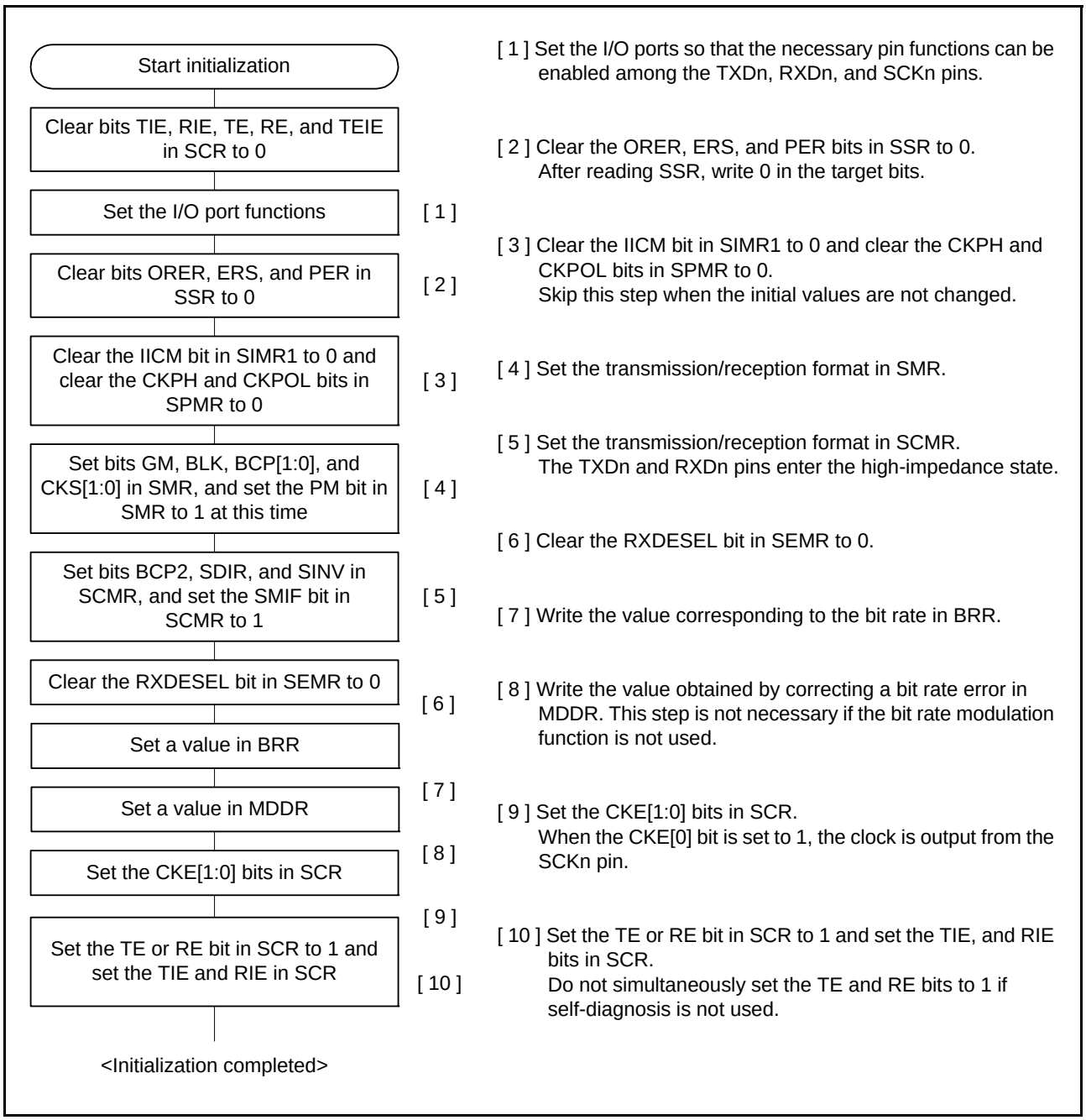



**Figure 26.37 Example of SCI Initialization Flowchart (Smart Card Interface Mode)**

### 26.6.6 Serial Data Transmission (Except in Block Transfer Mode)

Serial data transmission in smart card interface mode (except in block transfer mode), in that an error signal is sampled and data can be retransmitted, is different from that in non-smart card interface mode. Figure 26.38 shows the data retransfer operation during transmission.

1. When an error signal from the receiver end is sampled after one-frame data has been transmitted, the ERS flag in SSR is set to 1. If the RIE bit in SCR is 1 at this time, an ERI interrupt request is generated. Clear the ERS flag to 0 before the next parity bit is sampled.
2. For a frame in which an error signal is received, the TEND flag in SSR is not set. Data is retransferred from TDR to TSR allowing automatic data retransmission.
3. If no error signal is returned from the receiver, the ERS flag is not set to 1.
4. In this case, the SCI judges that transmission of one-frame data (including retransfer) has been completed, and the TEND flag is set. If the TIE bit in SCR is 1 at this time, a TXI interrupt request is generated. Writing transmit data to TDR starts transmission of the next data.

Figure 26.40 shows a sample flowchart of serial transmission. All the processing steps are automatically performed using a TXI interrupt request to activate the DTC.
When the TEND flag in SSR is set to 1 in transmission, if the TIE bit in SCR is 1, a TXI interrupt request is generated. The DTC is activated by a TXI interrupt request if the TXI interrupt request is specified as a source of DTC activation beforehand, allowing transfer of transmit data. The TEND flag is automatically cleared to 0 when the DTC transfers the data.
If an error occurs, the SCI automatically retransmits the same data. During this retransmission, the TEND flag is kept to 0 and the DTC is not activated. Therefore, the SCI and DTC automatically transmit the specified number of bytes, including retransmission in the case of error occurrence. However, since the ERS flag is not automatically cleared, set the RIE bit to 1 beforehand to enable an ERI interrupt request to be generated at error occurrence, and clear the ERS flag to 0.
When transmitting/receiving data using the DTC, be sure to make settings to enable the DTC before making SCI settings. For DTC settings, refer to section 16, Data Transfer Controller (DTCa).

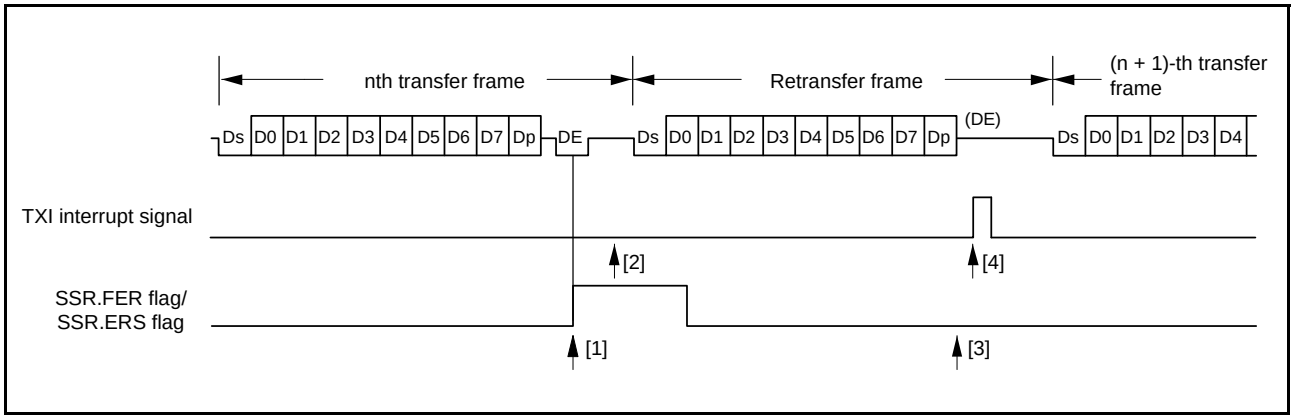



**Figure 26.38 Data Retransfer Operation in SCI Transmission Mode**

Note that the SSR.TEND flag is set in different timings depending on the GM bit setting in SMR. Figure 26.39 shows the TEND flag generation timing.

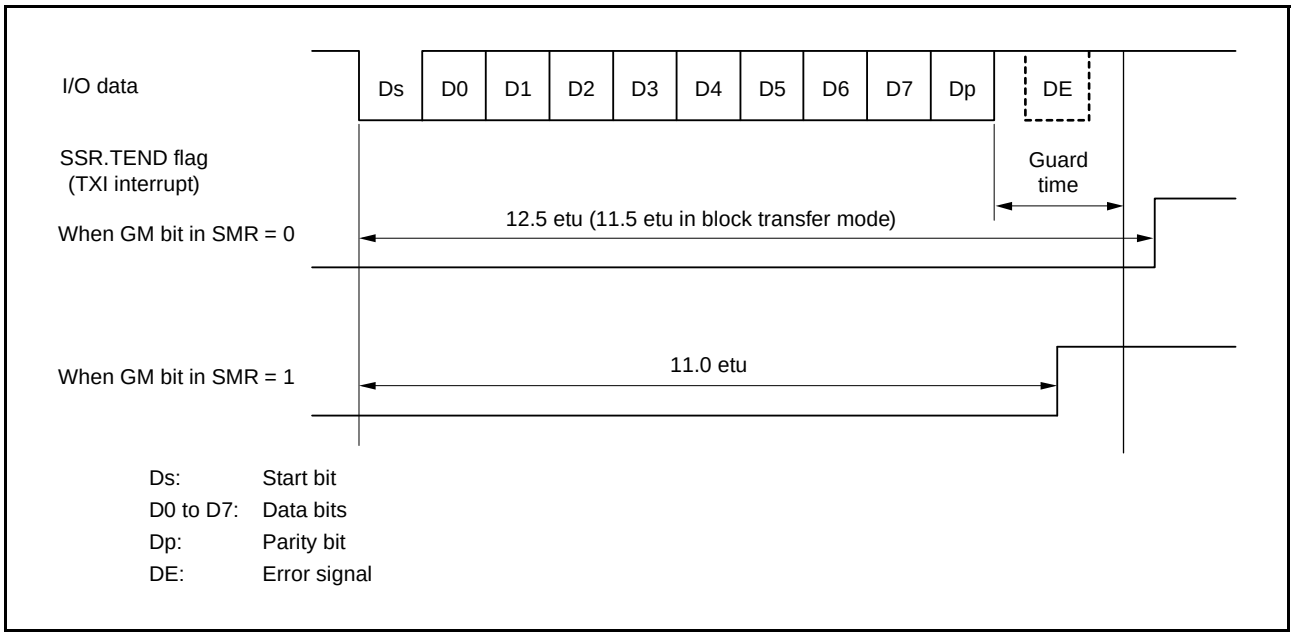



**Figure 26.39 SSR.TEND Flag Generation Timing during Transmission**

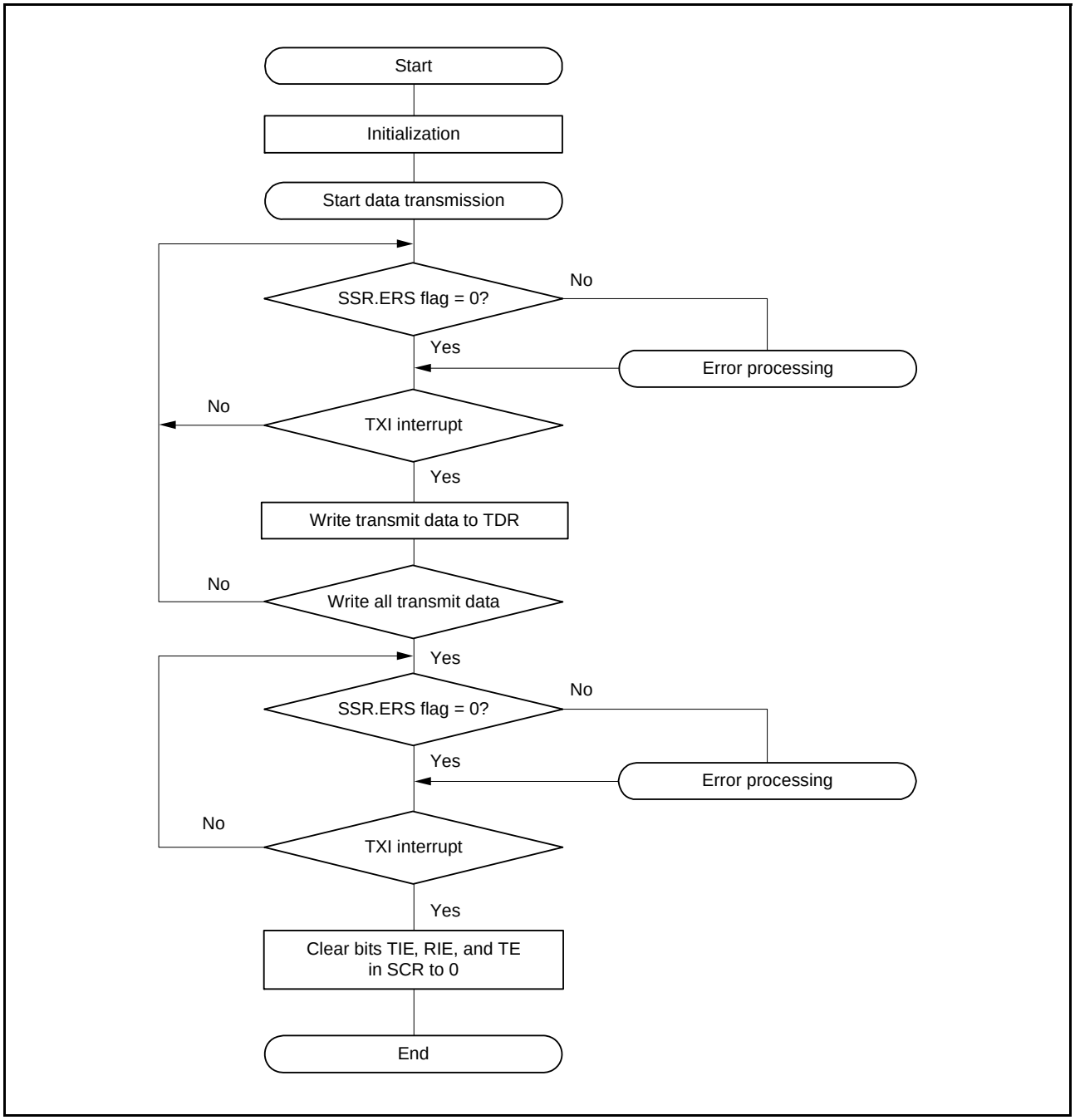



**Figure 26.40 Sample Smart Card Interface Transmission Flowchart**

### 26.6.7 Serial Data Reception (Except in Block Transfer Mode)

Serial data reception in smart card interface mode is similar to that in non-smart card interface mode. Figure 26.41 shows the data retransfer operation in reception mode.

1. If a parity error is detected in receive data, the PER flag in SSR is set to 1. When the RIE bit in SCR is 1 at this time, an ERI interrupt request is generated. Clear the PER flag to 0 before the next parity bit is sampled.
2. For a frame in which a parity error is detected, no RXI interrupt is generated.
3. When no parity error is detected, the PER flag in SSR is not set to 1.
4. In this case, data is determined to have been received successfully. When the RIE bit in SCR is 1, an RXI interrupt request is generated.

Figure 26.42 shows a sample flowchart for serial data reception. All the processing steps are automatically performed using an RXI interrupt request to activate the DTC.
In reception, setting the RIE bit to 1 allows an RXI interrupt request to be generated. The DTC is activated by an RXI interrupt request if the RXI interrupt request is specified as a source of DTC activation beforehand, allowing transfer of receive data.
If an error occurs during reception and either the ORER or PER flag in SSR is set to 1, a receive error interrupt (ERI) request is generated. Clear the error flag after the error occurrence. If an error occurs, the DTC is not activated and receive data is skipped. Therefore, the number of bytes of receive data specified in the DTC is transferred.
Even if a parity error occurs and the PER flag is set to 1 during reception, receive data is transferred to RDR, thus allowing the data to be read.
Note 1. For operations in block transfer mode, refer to section 26.3, Operation in Asynchronous Mode.

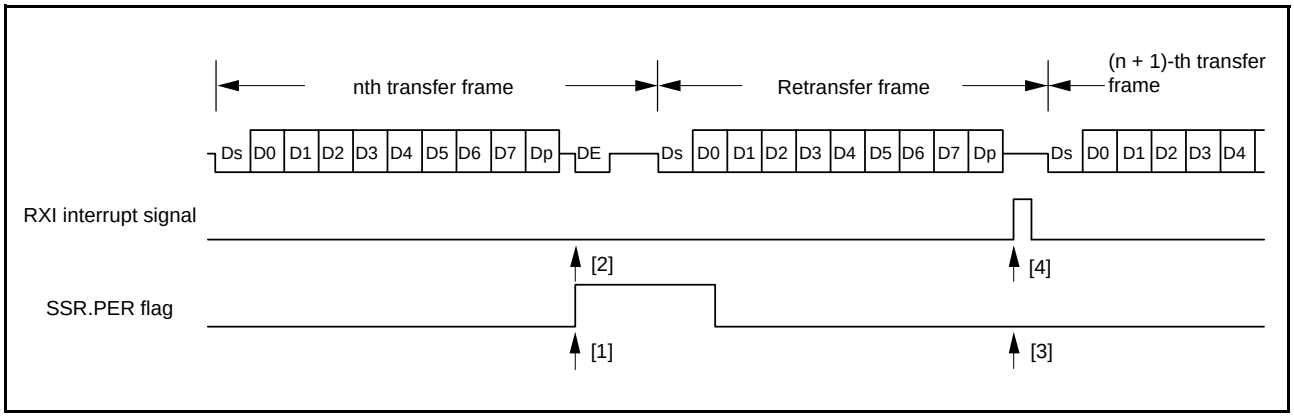



**Figure 26.41 Data Retransfer Operation in SCI Reception Mode (Data Retransfer Operation during Reception)**

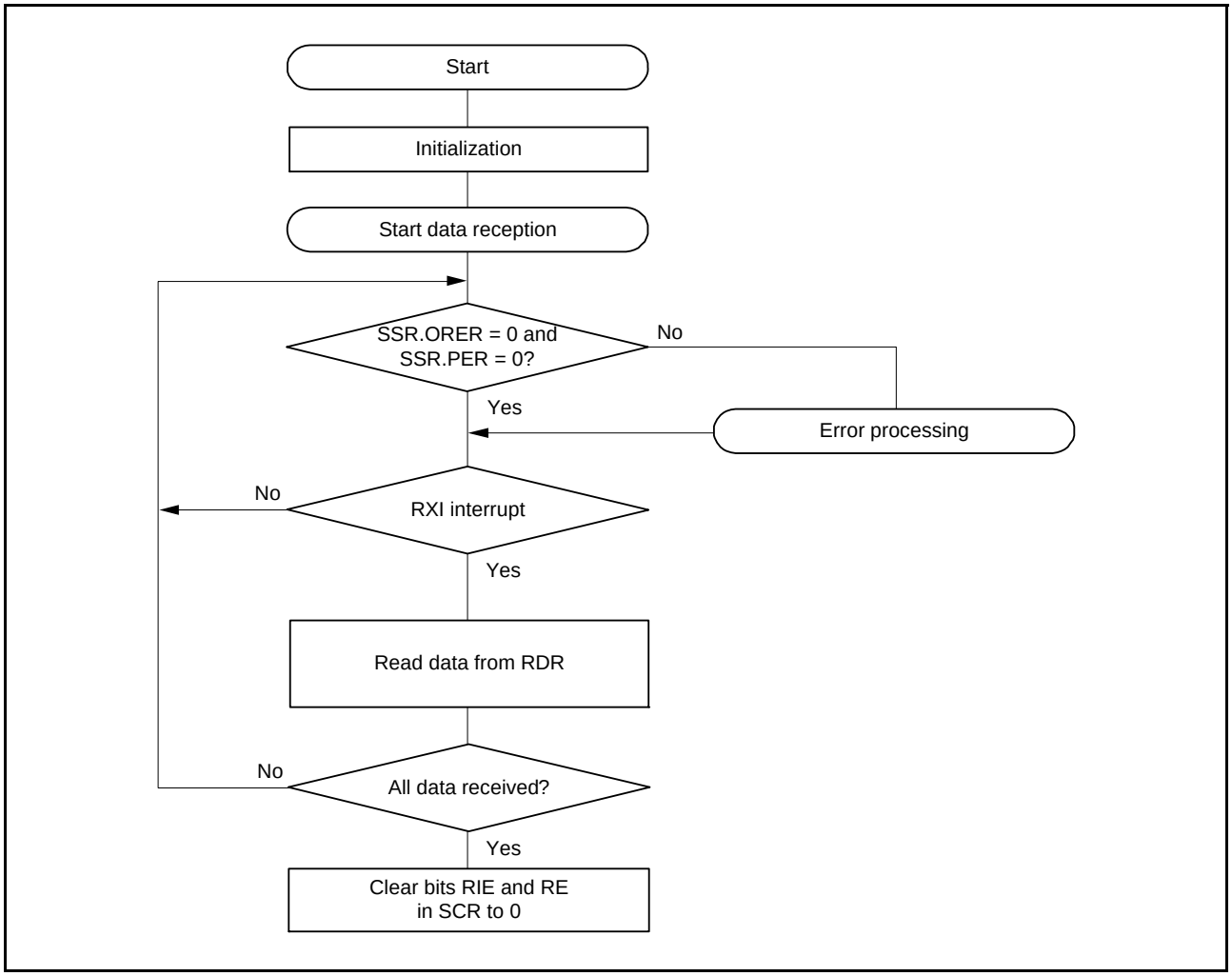



**Figure 26.42 Sample Smart Card Interface Reception Flowchart**

### 26.6.8 Clock Output Control

Clock output can be fixed using the CKE[1:0] bits in SCR when the GM bit in SMR is 1. Specifically, the minimum width of a clock pulse can be specified.

Figure 26.43 shows an example of clock output fixing timing when the CKE[0] bit is controlled with GM = 1 and CKE[1] = 0.

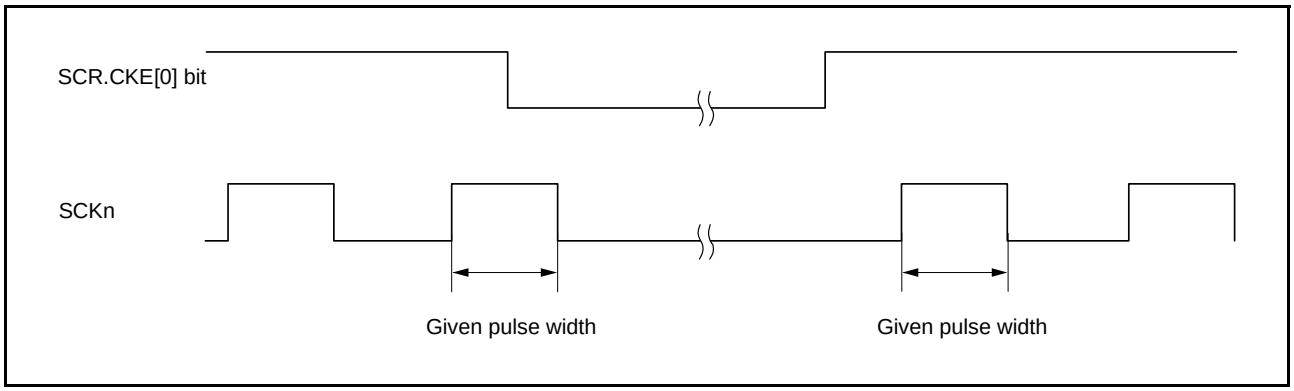



**Figure 26.43 Clock Output Fixing Timing**

At power-on and transitions to/from software standby mode, use the following procedure to secure the appropriate clock duty cycle.

#### (1) At Power-On

To secure the appropriate clock duty cycle simultaneously with power-on, use the following procedure.

1. Initially, port input is enabled in the high-impedance state. To fix the potential level, use a pull-up or pull-down resistor.
2. Fix the SCKn pin to the specified output by setting the SCR.CKE[1] bit and I/O port functions.
3. Set SMR and SCMR to enable smart card interface mode.
4. Set the SCR.CKE[0] bit to 1 to start clock output.

## 26.7 Operation in Simple I²C Mode

Simple I²C bus master format is composed of 8 data bits and an acknowledge bit. By continuing into a slave-address frame after a start condition or restart condition, a master device is able to specify a slave device as the partner for communications. The currently specified slave device remains valid until a new slave device is specified or a stop condition is satisfied. The 8 data bits in all frames are transmitted in order from the MSB.
The I²C format and timing of the I²C bus are shown in Figure 26.44 and Figure 26.45.

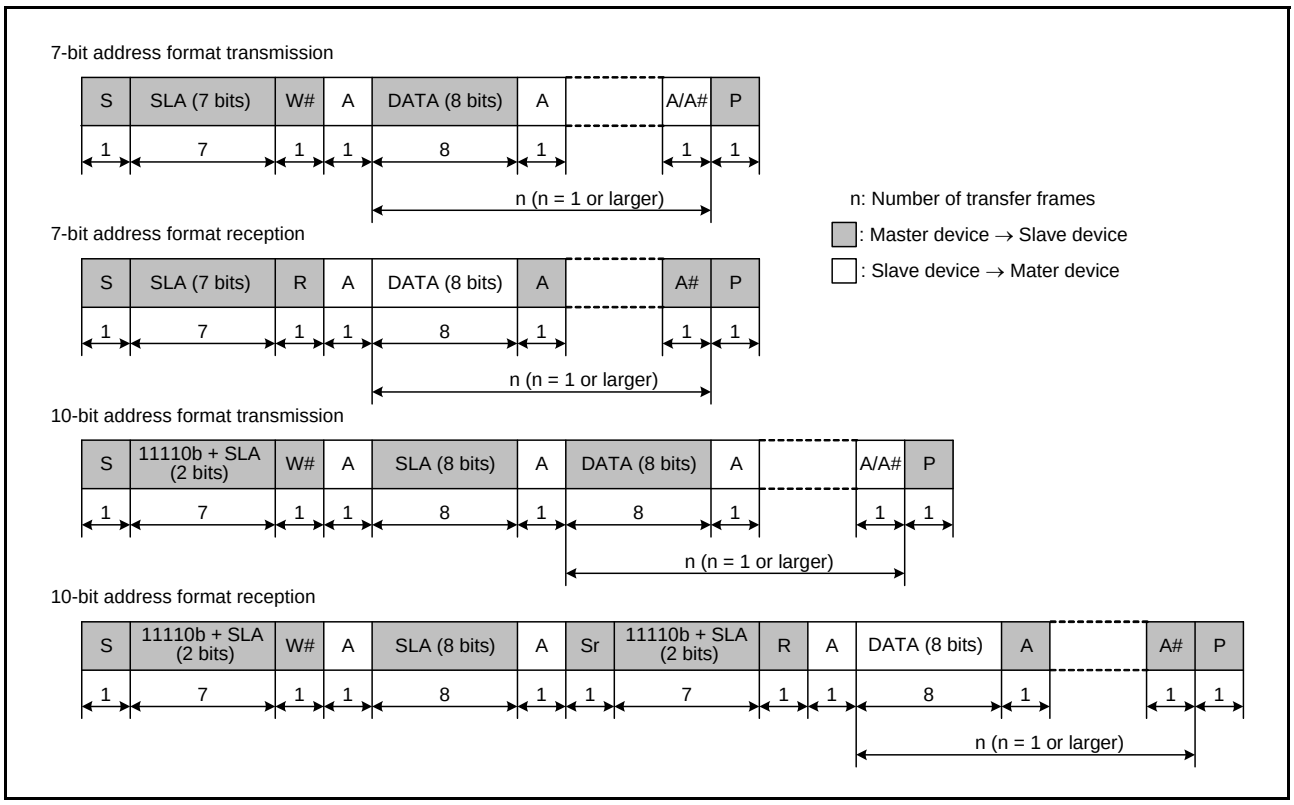



**Figure 26.44 I²C Bus Master Format**

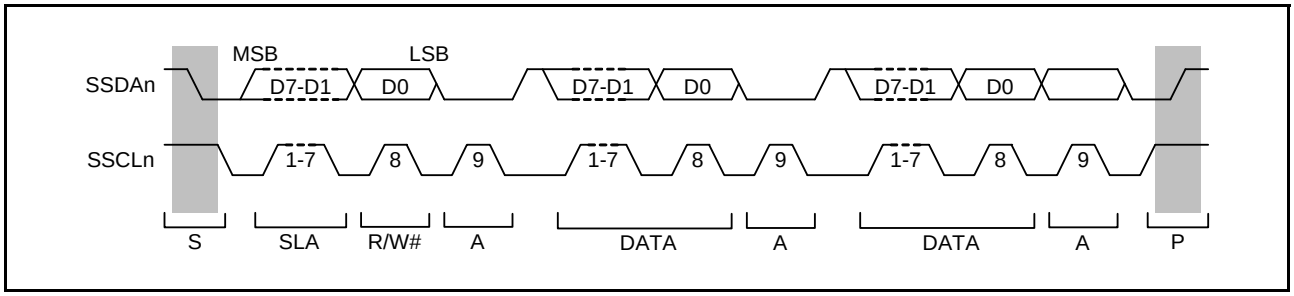



**Figure 26.45 I²C Bus Timing (When SLA is 7 Bits)**

S: Indicates a start condition, i.e. the master device changing the level on the SSDAn line from the high to the low level while the SSCLn line is at the high level.

SLA: Indicates a slave address, by which the master device selects a slave device.

R/W#: Indicates the direction of transfer (reception or transmission). The value 1 corresponds to transfer from the slave device to the master device and 0 corresponds to transfer from the master device to the slave device.

A/A#: Indicates an acknowledge bit. This is returned by the slave device for master transmission and by the master device for master reception. Return of the low level indicates ACK and return of the high level indicates NACK.

Sr: Indicates a restart condition, i.e. the master device changing the level on the SSDAn line from the high to the low level while the SSCLn line is at the high level and after the setup time has elapsed.

DATA: Indicates the data being received or transmitted.

P: Indicates a stop condition, i.e. the master device changing the level on the SSDAn line from the low to the high level while the SSCLn line is at the high level.

### 26.7.1 Generation of Start, Restart, and Stop Conditions

Writing 1 to the IICSTAREQ bit in SIMR3 causes the generation of a start condition. The generation of a start condition proceeds through the following operations.

- The level on the SSDAn line falls (from the high level to the low level) and the SSCLn line is kept in the released state.
- The hold time for the start condition is secured as half of a bit period at the bit rate determined by the setting of the BRR.
- The level on the SSCLn line falls (from the high level to the low level), the IICSTAREQ bit in SIMR3 is cleared (to 0), and a start-condition generated interrupt is output.

Writing 1 to the IICRSTAREQ bit in SIMR3 causes the generation of a start condition. The generation of a start condition proceeds through the following operations.

- The SSDAn line is released and the SSCLn line is kept at the low level.
- The period at low level for the SSCLn line is secured as half of a bit period at the bit rate determined by the setting of the BRR.
- The SSCLn line is released (transition from the low to the high level).
- Once the high level on the SSCLn line is detected, the setup time for the restart condition is secured as half of a bit period at the bit rate determined by the setting of the BRR.
- The level on the SSDAn line falls (from the high level to the low level).
- The hold time for the restart condition is secured as half of a bit period at the bit rate determined by the setting of the BRR.
- The level on the SSCLn line falls (from the high level to the low level), the IICRSTAREQ bit in SIMR3 is cleared (to 0), and a restart-condition generated interrupt is output.

Writing 1 to the IICSTPREQ bit in SIMR3 causes the generation of a stop condition. The generation of a stop condition proceeds through the following operations.

- The level on the SSDAn line falls (from the high level to the low level) and the SSCLn line is kept at the low level.
- The period at low level for the SSCLn line is secured as half of a bit period at the bit rate determined by the setting of the BRR.
- The SSCLn line is released (transition from the low to the high level).
- Once the high level on the SSCLn line is detected, the setup time for the stop condition is secured as half of a bit period at the bit rate determined by the setting of the BRR.
- The SSDAn is released (transition from the low to the high level), the IICSTPREQ bit in SIMR3 is cleared (to 0), and a stop-condition generated interrupt is output.

Figure 26.46 shows the timing of operations in the generation of start, restart, and stop conditions.

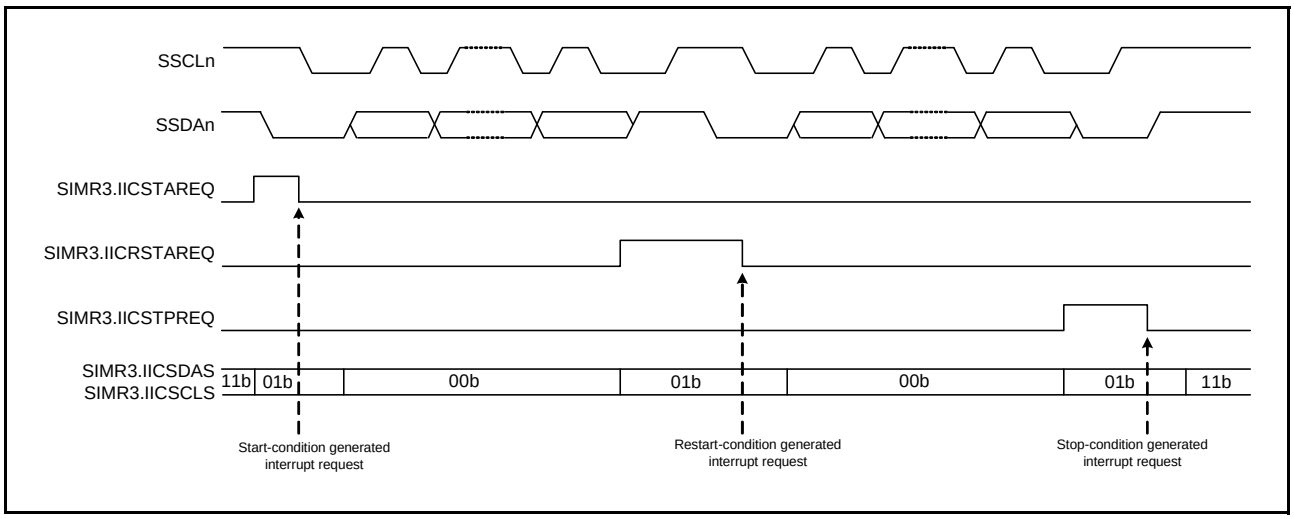



**Figure 26.46 Timing of Operations in the Generation of Start, Restart, and Stop Conditions**

### 26.7.2 Clock Synchronization

The SSCLn line may be placed at the low level in the case of a wait inserted by a slave device as the other side of transfer. Setting the IICCSC bit in SIMR2 to 1 applies control to obtain synchronization when the levels of the internal SSCLn clock signal and the level being input on the SSCLn pin differ.

When the IICCSC bit in SIMR2 is set to 1, the level of the internal SSCLn clock signal changes from low to high, counting to determine the period at high level is stopped while the low level is being input on the SSCLn pin, and counting to determine the period at high level starts after the transition of the input on the SSCLn pin to the high level. The interval from this time until counting to determine the period at high level starts on the transition of the SSCLn pin to the high level is the total of the delay of SSCLn output, delay for noise filtering of the input on the SSCLn pin (2 or 3 cycles of sampling clock for the noise filter), and delay for internal processing (1 or 2 cycles of PCLK). The period at high level of the internal SSCLn clock is extended even if other devices are not placing the low level on the SSCLn line.

If the IICCSC bit in SIMR2 is 1, synchronization is obtained for the transmission and reception of data by taking the logical AND of the input on the SSCLn pin and the internal SSCLn clock. If the IICCSC bit in SIMR2 is 0, synchronization with the internal SSCLn clock is obtained for the transmission and reception of data.

If a slave device inserts a period of waiting into the interval until the transition of the internal SSCLn clock signal from the low to the high level after a request for the generation of a start, restart, or stop condition is issued, the time until generation is prolonged by that period.

If a slave device inserts a period of waiting after the transition of the internal SSCLn clock signal from the low to the high level, although the generation-completed interrupt is issued without stopping the period of waiting, generation of the condition itself is not guaranteed. Figure 26.47 shows an example of operations to synchronize the clocks.

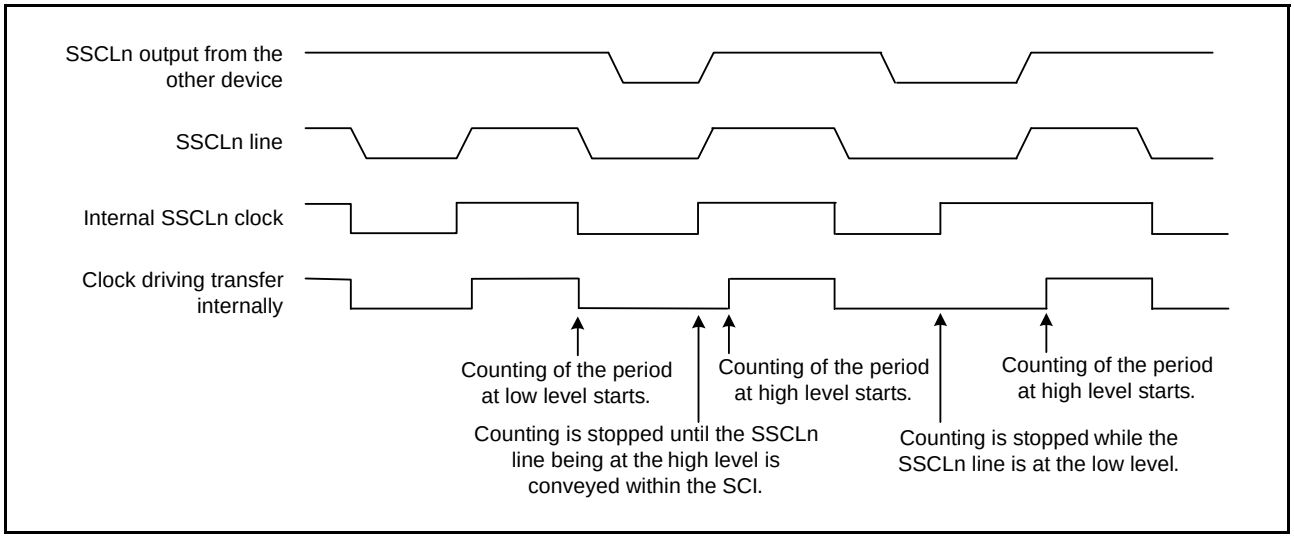



**Figure 26.47 Example of Operations for Clock Synchronization**

### 26.7.3 SSDA Output Delay

The IICDL[4:0] bits in SIMR1 can be used to set a delay for output on the SSDAn pin relative to falling edges of output on the SSCLn pin. Delay-time settings from 0 to 31 are selectable, representing periods of the corresponding numbers of cycles of the clock signal from the on-chip baud rate generator (derived by frequency-dividing the base clock, PCLK, by the divisor selected by the CKS[1:0] bits in SMR). A delay for output on the SSDAn pin is for the start condition/restart condition/stop condition signal, 8-bit transmit data, and an acknowledge bit.

If the SSDA output delay is shorter than the time for the level on the SSCLn pin to fall, the change of the output on the SSDAn pin will start while the output level on the SSCLn pin is falling, creating a possibility of erroneous operation for slave devices. Ensure that settings for the delay of output on the SSDAn pin are for times greater than the time output on the SSCLn pin takes to fall (300 ns for $I^2C$ in normal mode and fast mode).

Figure 26.48 shows the timing of delays in SSDA output.

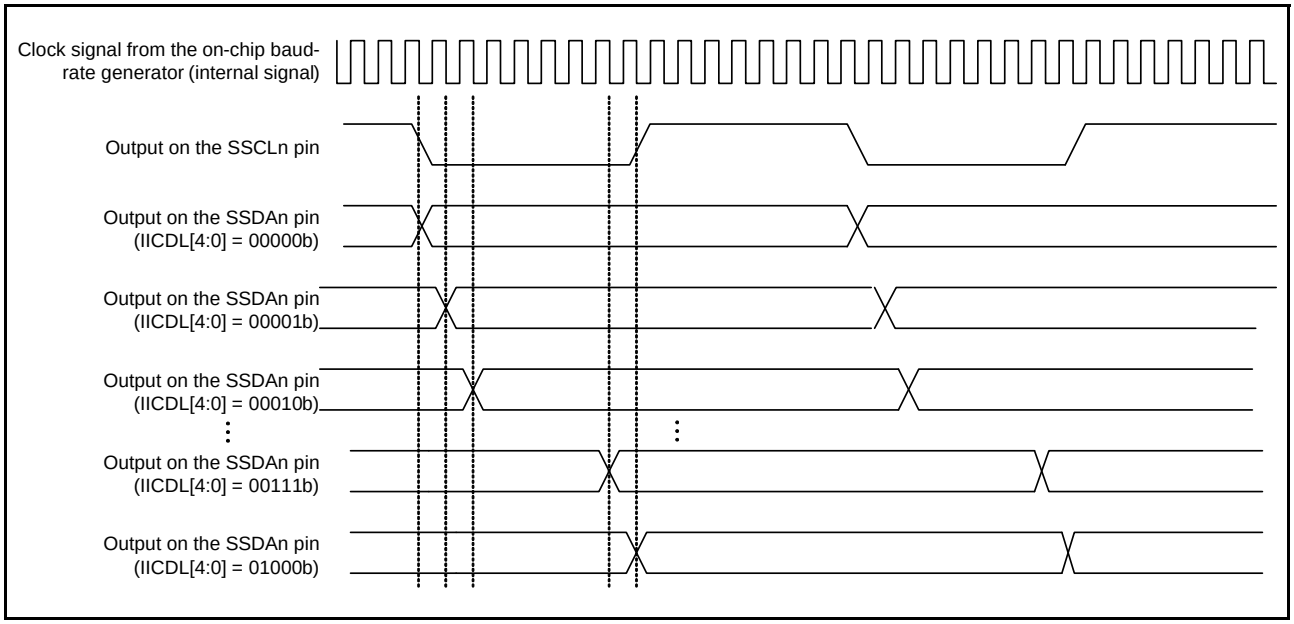


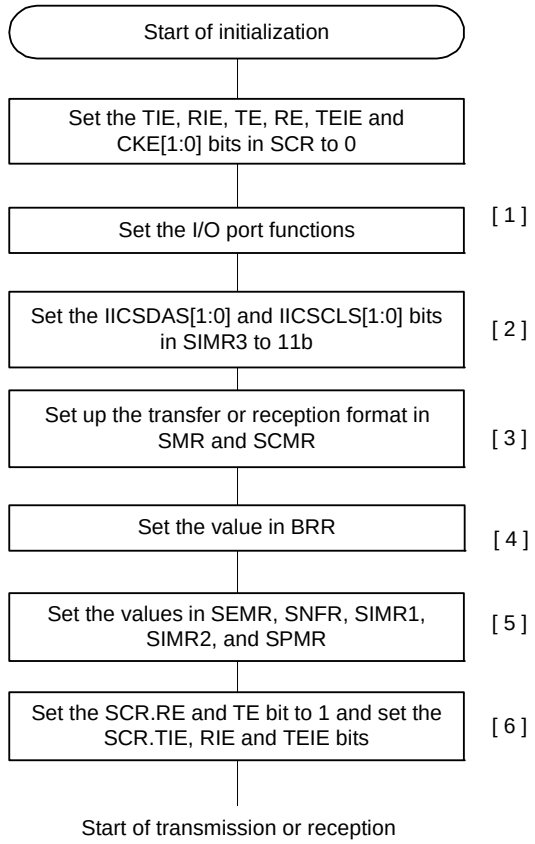

**Figure 26.48 Timing of Delays in SSDA Output**

## 26.7.4 SCI Initialization (Simple I2C Mode)

Before transferring data, write the initial value (00h) to SCR and initialize the interface in accordance with the flowchart shown as Figure 26.49.

When changing the operating mode, transfer format, and so on, be sure to set SCR to its initial value before proceeding with the changes.

In simple I2C mode, the open drain setting for the communication ports should be made on the port side.




[ 1 ] Make I/O port settings that allow use (on N-channel open-drain output pins) of the SSCLn and SSDAn pin functions.

[ 2 ] Place the SSCLn and SSDAn pins in the high-impedance state until a start condition is to be generated.

[ 3 ] Set the format for transmission and reception in SMR and SCMR.
In SMR, set the CKS[1:0] bits to the desired value and set the other bits to 0.
In SCMR, set the SDIR bit to 1 and the SINV and SMIF bits to 0.

[ 4 ] Write the value for the desired bit rate to BRR.

[ 5 ] Set the values in SEMR, SNFR, SIMR1, SIMR2, and SPMR.
In SEMR, set the NFEN bit.
In SNFR, set the NFCS[2:0] bits.
In SIMR1, set the IICM bit to 1 and the IICDL[4:0] bits as required.
In SIMR2, set the IICACKT and IICCSC bits to 1 and the IICINTM bits as required.
In SPMR, clear all the bits to 0.

[ 6 ] Set the RE and TE bits in the SCR to 1. Then, set the SCR.TIE, RIE, and TEIE bits (for transmission and when the SIMR2.IICINTM bit is 1, clear the RIE bit). Setting the TE and RE bits to 1 makes the SSCLn and SSDAn pin functions available.

Note: • Set the SCR.RE and TE bits to 0 or 1 at the same time.

**Figure 26.49 Example of the Flow of SCI Initialization (for Simple I2C Mode)**

## 26.7.5 Operation in Master Transmission (Simple I2C Mode)

Figure 26.50 and Figure 26.51 show examples of operations in master transmission and Figure 26.52 is a flowchart showing the procedure for data transmission. The value of the SIMR2.IICINTM bit is assumed to be 1 (transmission and reception interrupts are in use) and the value of the SCR.RIE bit is assumed to be 0 (disabling reception interrupt requests). See Table 26.30 for more information on the STI interrupt.
When 10-bit slave addresses are in use, steps [3] and [4] in Figure 26.52 are repeated twice.
In simple I2C mode, the transmit data empty interrupt (TXI) is generated when communication of one frame is completed, unlike the TXI interrupt request generation timing during clock synchronous transmission.

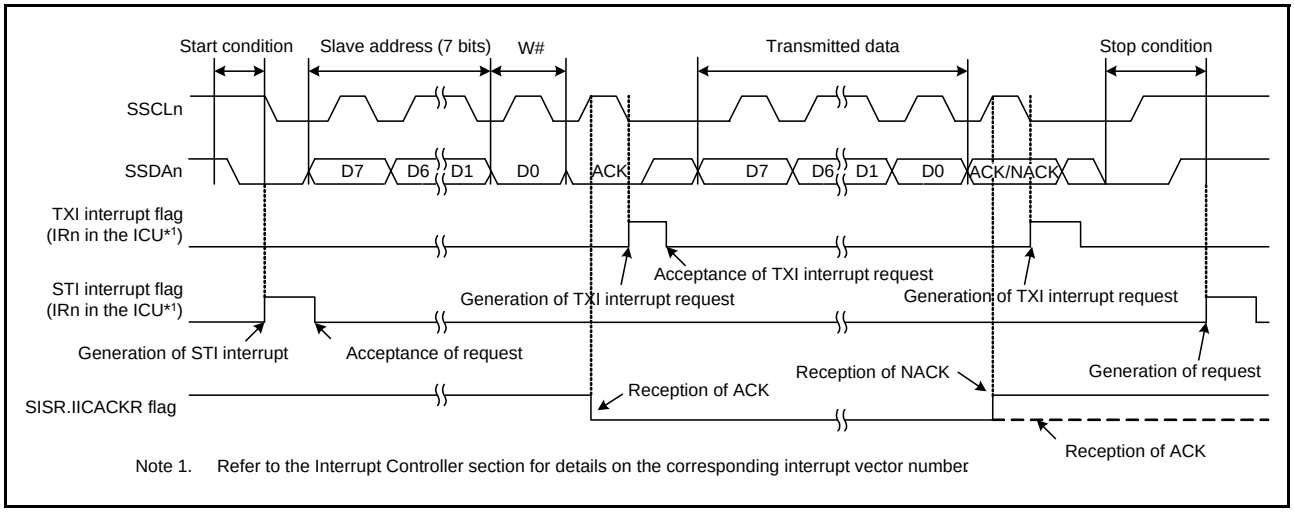



**Figure 26.50 Example 1 of Operations for Master Transmission in Simple I2C Bus Mode (with 7-Bit Slave Addresses, Transmission Interrupts, and Reception Interrupts in Use)**

When the SIMR2.IICINTM bit is set to 0 (ACK and NACK interrupts are used) during master transmission, the DTC is activated by the ACK interrupt as the trigger and necessary number of data bytes are transmitted. When the NACK is received, error processing, such as transmission stop and retransmission, is performed by the NACK interrupt as the trigger.

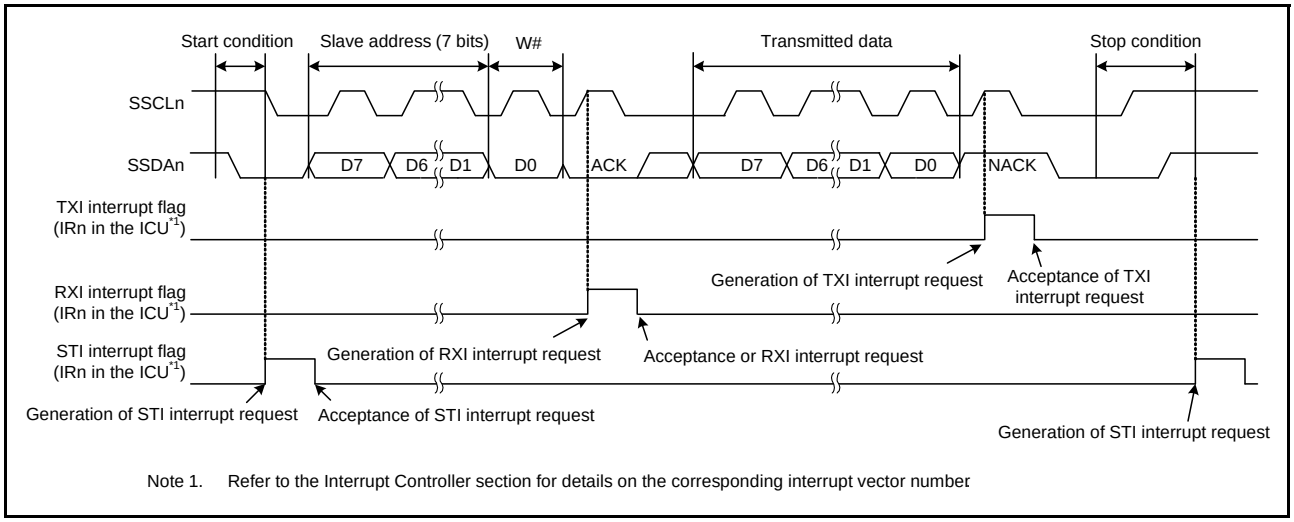
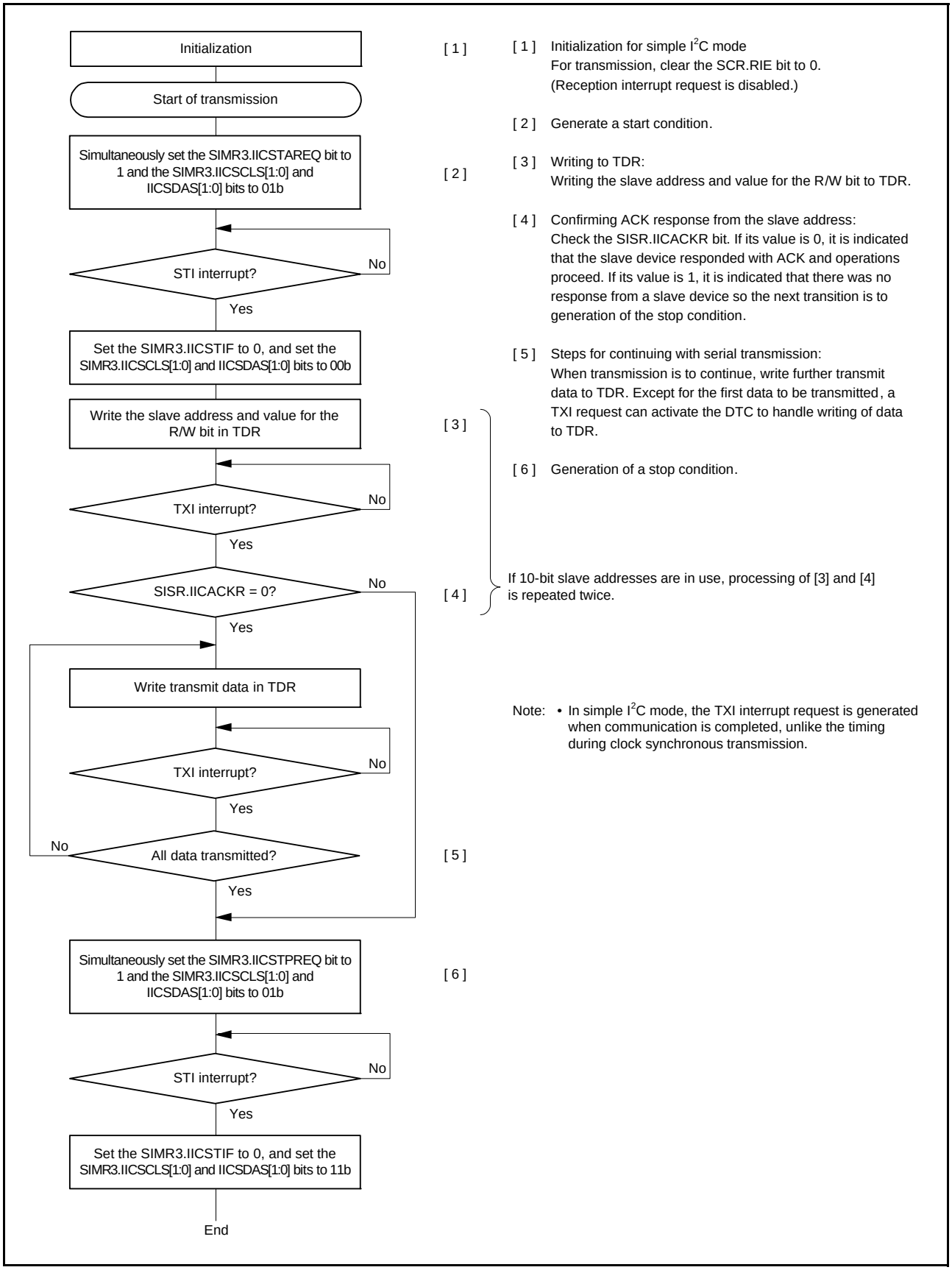



**Figure 26.51 Example 2 of Operations for Master Transmission in Simple I2C Bus Mode (with 7-Bit Slave Addresses, ACK Interrupts, and NACK Interrupts in Use)**




[ 1 ] Initialization for simple I²C mode
For transmission, clear the SCR.RIE bit to 0.
(Reception interrupt request is disabled.)

[ 2 ] Generate a start condition.

[ 3 ] Writing to TDR:
Writing the slave address and value for the R/W bit to TDR.

[ 4 ] Confirming ACK response from the slave address:
Check the SISR.IICACKR bit. If its value is 0, it is indicated that the slave device responded with ACK and operations proceed. If its value is 1, it is indicated that there was no response from a slave device so the next transition is to generation of the stop condition.

[ 5 ] Steps for continuing with serial transmission:
When transmission is to continue, write further transmit data to TDR. Except for the first data to be transmitted, a TXI request can activate the DTC to handle writing of data to TDR.

[ 6 ] Generation of a stop condition.

If 10-bit slave addresses are in use, processing of [3] and [4] is repeated twice.

Note: • In simple I²C mode, the TXI interrupt request is generated when communication is completed, unlike the timing during clock synchronous transmission.

**Figure 26.52 Example of the Procedure for Master Transmission Operations in Simple I²C Mode (with Transmission Interrupts and Reception Interrupts in Use)**

## 26.7.6 Master Reception (Simple I2C Mode)

Figure 26.53 shows an example of operations in simple I2C mode master reception and Figure 26.54 is a flowchart showing the procedure for master reception.
The value of the SIMR2.IICINTM bit is assumed to be 1 (transmission and reception interrupts are in use).
In simple I2C mode, the transmit data empty interrupt (TXI) is generated when communication of one frame is completed, unlike the TXI interrupt request generation timing during clock synchronous transmission.

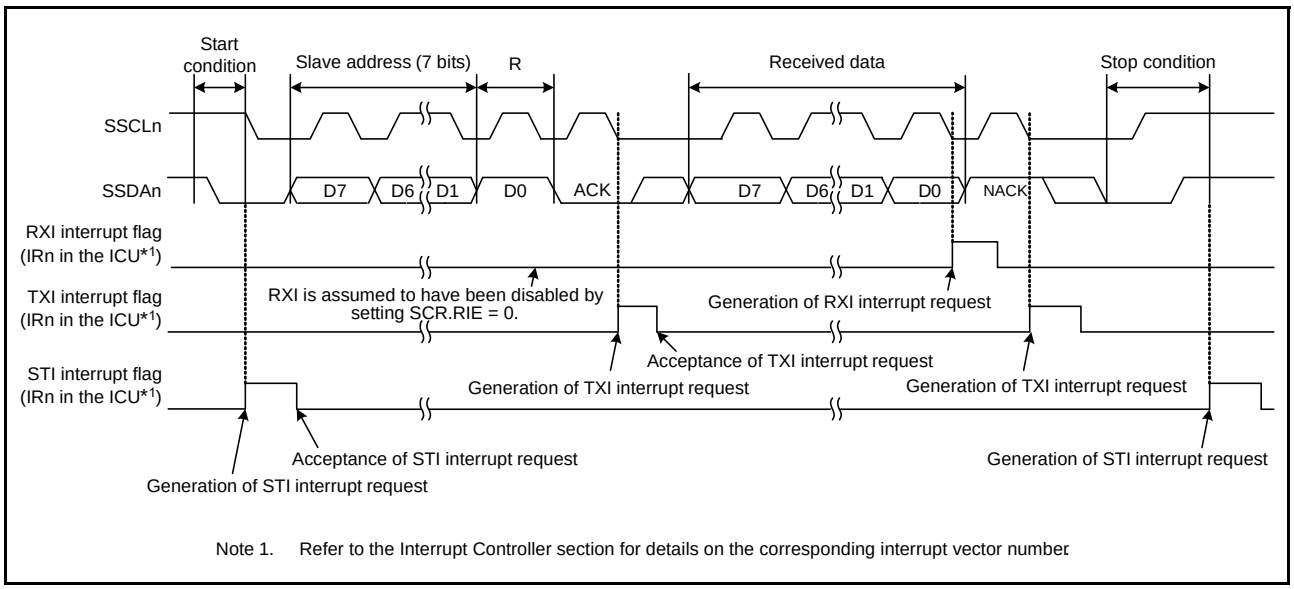
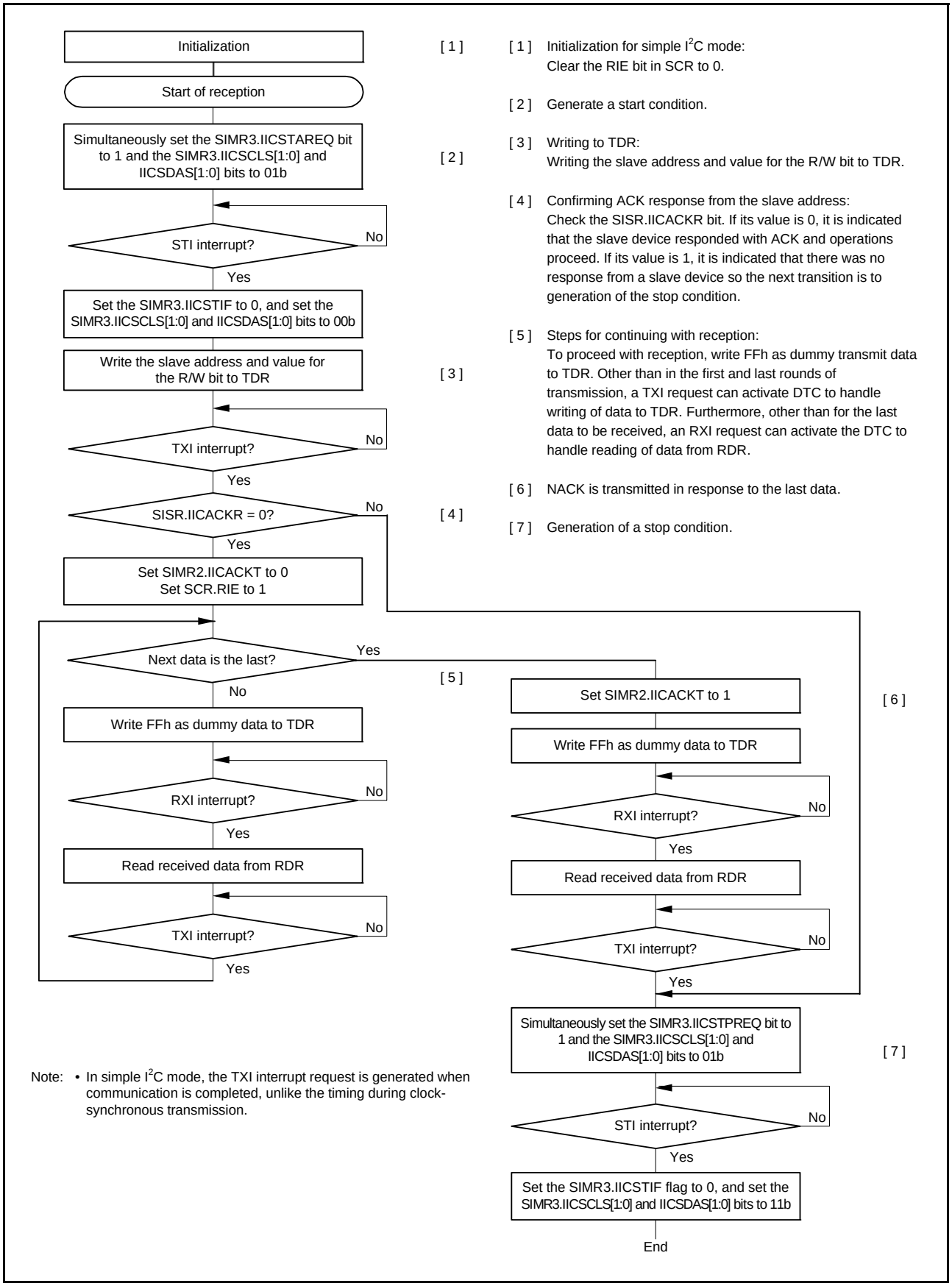



**Figure 26.53 Example of Operations for Master Reception in Simple I2C Bus Mode (with 7-Bit Slave Addresses, Transmission Interrupts, and Reception Interrupts in Use)**




[ 1 ] Initialization for simple I²C mode:
Clear the RIE bit in SCR to 0.

[ 2 ] Generate a start condition.

[ 3 ] Writing to TDR:
Writing the slave address and value for the R/W bit to TDR.

[ 4 ] Confirming ACK response from the slave address:
Check the SISR.IICACKR bit. If its value is 0, it is indicated that the slave device responded with ACK and operations proceed. If its value is 1, it is indicated that there was no response from a slave device so the next transition is to generation of the stop condition.

[ 5 ] Steps for continuing with reception:
To proceed with reception, write FFh as dummy transmit data to TDR. Other than in the first and last rounds of transmission, a TXI request can activate DTC to handle writing of data to TDR. Furthermore, other than for the last data to be received, an RXI request can activate the DTC to handle reading of data from RDR.

[ 6 ] NACK is transmitted in response to the last data.

[ 7 ] Generation of a stop condition.

Note: • In simple I²C mode, the TXI interrupt request is generated when communication is completed, unlike the timing during clock-synchronous transmission.

**Figure 26.54 Example of the Procedure for Master Reception Operations in Simple I²C Mode (with Transmission Interrupts and Reception Interrupts in Use)**

## 26.8 Operation in Simple SPI Mode

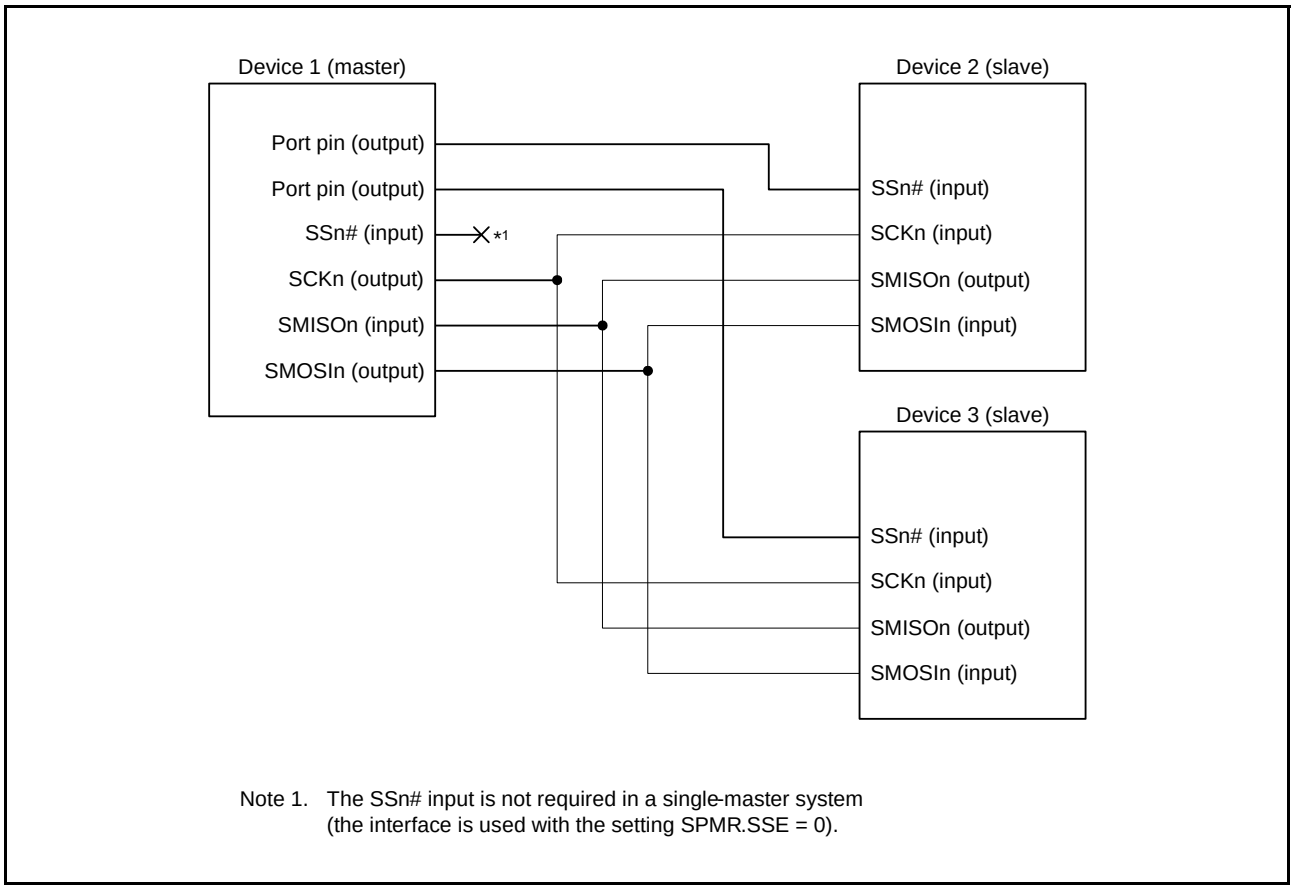
As an extended function, the SCI supports a simple SPI mode that handles transfer among one or multiple master devices and multiple slave devices.

Making the settings for clock synchronous mode (SCMR.SMIF = 0, SIMR1.IICM = 0, SMR.CM = 1) plus setting the SSE bit in the SPMR to 1 places the SCI in simple SPI mode. However, the SS pin function on the master side is unnecessary for connection of the device used as the master in simple SPI mode when the configuration only has a single master, so set the SSE bit in the SPMR to 0 in such cases.

Figure 26.55 shows an example of connections for simple SPI mode. Control a general port pin to produce the SS output signal from the master.

In simple SPI mode, data are transferred in synchronization with clock pulses in the same way as in clock synchronous mode. One character of data for transfer consists of 8 bits of data, and parity bits cannot be appended to this. The data can be inverted by setting the SCMR.SINV bit to 1.

Since the receiver and transmitter are independent of each other within the SCI module, full-duplex communications are possible, with a common clock signal. Furthermore, since both the transmitter and receiver have a buffered structure, writing of further transmit data while transmission is in progress and reading of previously received data while reception is in progress are both possible. Continuous transfer is thus possible.




Note 1. The SSn# input is not required in a single-master system (the interface is used with the setting SPMR.SSE = 0).

**Figure 26.55 Example of Connections via a Simple SPI Mode (In Single Master Mode, SPMR.SSE Bit = 0)**

### 26.8.1 States of Pins in Master and Slave Modes

The direction (input or output) of pins for the simple SPI mode interface differs according to whether the device is a master (SCR.CKE[1:0] = 00b or 01b and SPMR.MSS = 0) or slave (SCR.CKE[1:0] = 10b or 11b and SPMR.MSS = 1). Table 26.26 lists the states of pins according to the mode and the level on the SSn# pin.

**Table 26.26 States of Pins by Mode and Input Level on the SSn# Pin**

| Mode | Input on SSn# Pin | State of SMOSIn Pin | State of SMISOn Pin | State of SCKn Pin |
|---|---|---|---|---|
| Master mode*1 | High level (transfer can proceed) | Output for data transmission*2 | Input for received data | Clock output*3 |
| | Low level (transfer cannot proceed) | High-impedance | Input for received data (but disabled) | High-impedance |
| Slave mode | High level (transfer can proceed) | Input for received data (but disabled) | High-impedance | Clock input (but disabled) |
| | Low level (transfer cannot proceed) | Input for received data | Output for data transmission | Clock input |

Note 1. When there is only a single master (SPMR.SSE = 0), transfer is possible regardless of the input level on the SSn# pin (this is equivalent to input of a high level on the SSn# pin). Since the SSn# pin function is not required, the pin is available for other purposes.
Note 2. The SMOSIn pin output is in the high-impedance state when serial transmission is disabled (SCR.TE bit = 0).
Note 3. The SCKn pin output is in the high-impedance state when serial transmission is disabled (SCR.TE and RE bits = 00b) in a multi-master configuration (SPMR.SSE = 1).

### 26.8.2 SS Function in Master Mode

Setting the CKE[1:0] bits in the SCR to 00b and the MSS bit in the SPMR to 0 selects master operation. The SSn# pin is not used in single-master configurations (SPMR.SSE = 0), so transmission or reception can proceed regardless of the value of the SSn# pin.

When the level on the SSn# pin is high in a multi-master configuration (SPMR.SSE = 1), a master device outputs clock signals from the SCKn pin before starting transmission or reception to indicate that there are no other masters or another master is performing reception or transmission. When the level on the SSn# pin is low in a multi-master configuration (SPMR.SSE = 1), there are other masters, and this indicates that transmission or reception is in progress. At this time the SMOSIn output and SCKn pins will be placed in the high-impedance state and starting transmission or reception will not be possible. Furthermore, the value of the SPMR.MFF bit will be 1, indicating a mode-fault error. In a multi-master configuration, start error processing by reading SPMR.MFF flag. Also, even if a mode-fault error occurs while transmission or reception is in progress, transmission or reception will not be stopped, but the SMOSIn and SCKn pin output will be placed in the high-impedance state after the completion of the transfer.

Control a general port pin to produce the SS output signal from the master.

### 26.8.3 SS Function in Slave Mode

Setting the CKE[1:0] bits in the SCR to 10 and the MSS bit in the SPMR to 1 selects slave operation. When the level on the SSn# pin is high, the SMISOn output pin will be in the high-impedance state and clock input through the SCKn pin will be ignored. When the level on the SSn# pin is low, clock input through the SCKn pin will be effective and transmission or reception can proceed.

If the input on the SSn# pin changes from low to high level during transmission or reception, the SMISOn output pin will be placed in the high-impedance state. Meanwhile, the internal processing for transmission or reception will continue at the rate of the clock input through the SCKn pin until processing for the character currently being transmitted or received is completed, after which it stops. At that time, an interrupt (the appropriate one from among TXI, RXI, and TEI) will be generated.

### 26.8.4 Relationship between Clock and Transmit/Receive Data

The CKPOL and CKPH bits in the SPMR can be used to set up the clock for use in transmission and reception in four different ways. The relation between the clock signal and the transmission and reception of data is shown in Figure 26.56. The relation is the same for both master and slave operation. This is the same as when the level on the SSn# pin is high. The SSn# pin can be used for another purpose. For details, refer to section 26.8.2, SS Function in Master Mode.

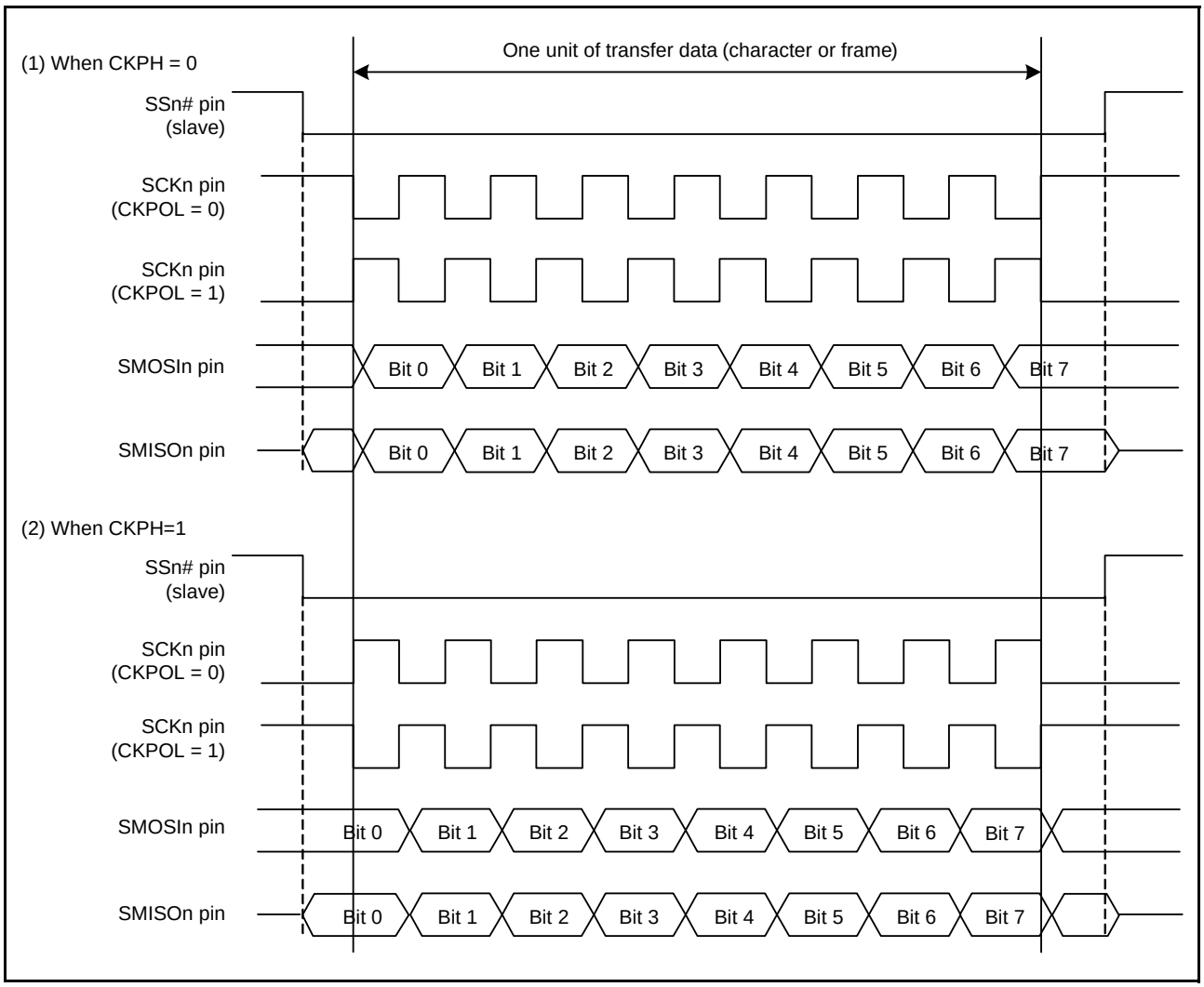



**Figure 26.56 Relation between Clock Signal and Transmit/Receive Data in Simple SPI Mode**

### 26.8.5 SCI Initialization (Simple SPI Mode)

The procedure is the same as for initialization in clock synchronous mode Figure 26.23, Sample SCI Initialization Flowchart. The CKPOL and CKPH bits in the SPMR must be set to ensure that the kind of clock signal they select is suitable for both master and slave devices.
For initialization, changes to the operating mode, changes to the transfer format, and so on, initialize the SCR register before proceeding with changes.
As well as setting the RE bit to 0, note that the SSR.ORER, FER, and PER flags, as well as the RDR, are not initialized. Note that changing the value of the TE bit from 1 to 0 or from 0 to 1 will lead to the generation of a transmit data empty interrupt (TXI) if the value of the TIE bit in the SCR is 1 at the time.

### 26.8.6 Transmission and Reception of Serial Data (Simple SPI Mode)

In master operation, ensure that the SSn# pin of the slave device on the other side of the transfer is at the low level before starting the transfer and at the high level on completion of the transfer. Otherwise, the procedures are the same as in clock synchronous mode.

## 26.9 Extended Serial Mode Control Section: Description of Operation

### 26.9.1 Serial Transfer Protocol

In conjunction with an SCIe module, the extended serial mode control section of an SCIf module can realize the serial transfer protocol composed of Start Frames and Information Frames that is shown in Figure 26.57.
A Start Frame is composed of a Break Field, Control Field 0, and Control Field 1. An Information Frame is composed of a number of Data Fields, a CRC16 Upper Field, and a CRC16 Lower Field.

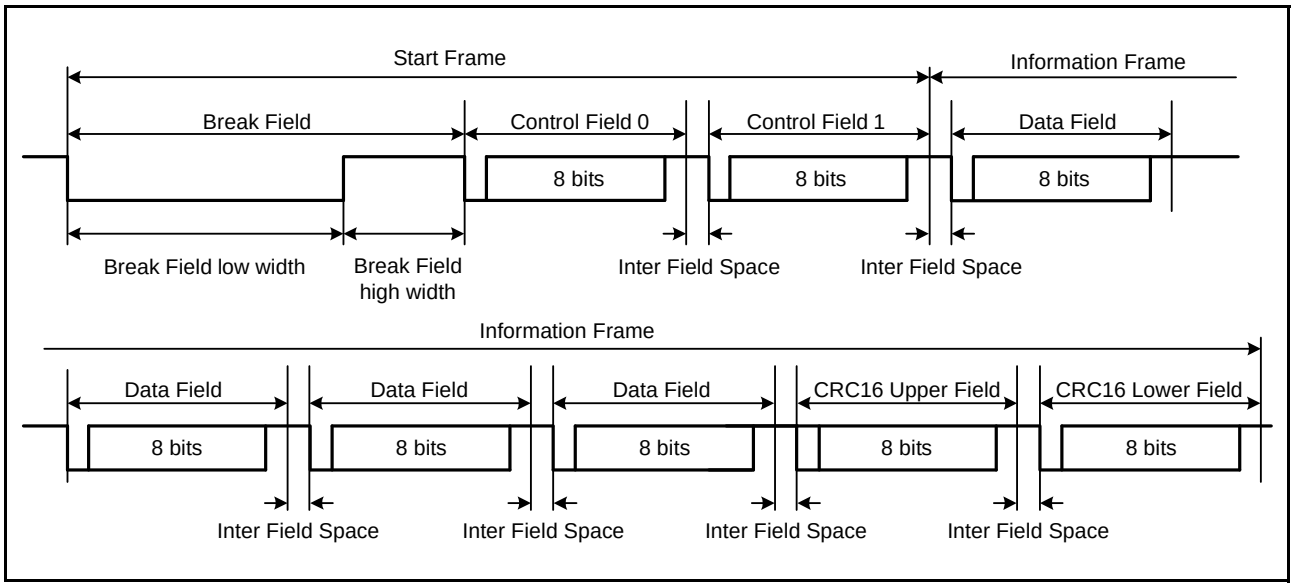



**Figure 26.57 Protocol for Serial Transfer by the Extended Serial Mode Control Section**

### 26.9.2 Transmitting a Start Frame

Figure 26.58 shows an example of operations to transmit a Start Frame, which is composed of the Break Field low width, Control Field 0, and Control Field 1. Figure 26.59 and Figure 26.60 are flowcharts for the transmission of a Start Frame.
Operations when the extended serial mode control section is to be used to transmit a Start Frame are as listed below. Be sure to use the SCI12 in asynchronous mode.

(1) With Break Field low width output mode as the operating mode for the timer, writing 1 to the TCST bit in TCR starts counting by the timer, and the low level will be output from the TXDX12 pin over the period corresponding to the TCNT and TPRE settings.
(2) The output on the TXDX12 pin is inverted when the timer counter underflows, and the BFDF bit in STR is set to 1. An SCIX0 interrupt is also generated if the value of the BFDIE bit in ICR is 1.
(3) Writing 0 to the TCST bit in TCR stops counting by the timer, and SCI2 is used to send the data for Control Field 0. After the Break Field low width output, stop counting before the next underflow occurs.
(4) Once the data for Control Field 0 have been transmitted, SCI12 is used to send the data for Control Field 1.
(5) Once the data for Control Field 1 have been transmitted, SCI12 is used to send an Information Frame.

Omit the Break Field and Control Field 0 to suit the structure of the Start Frame.

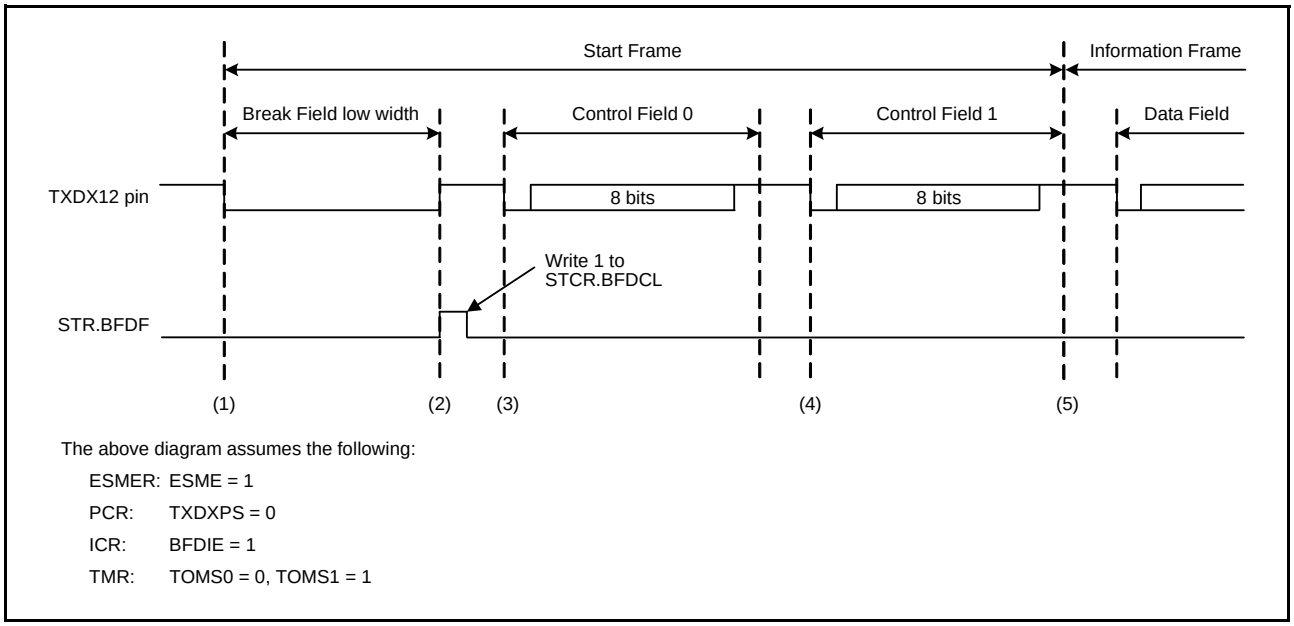



**Figure 26.58 Example of Operations at the Time of Start Frame Transmission**

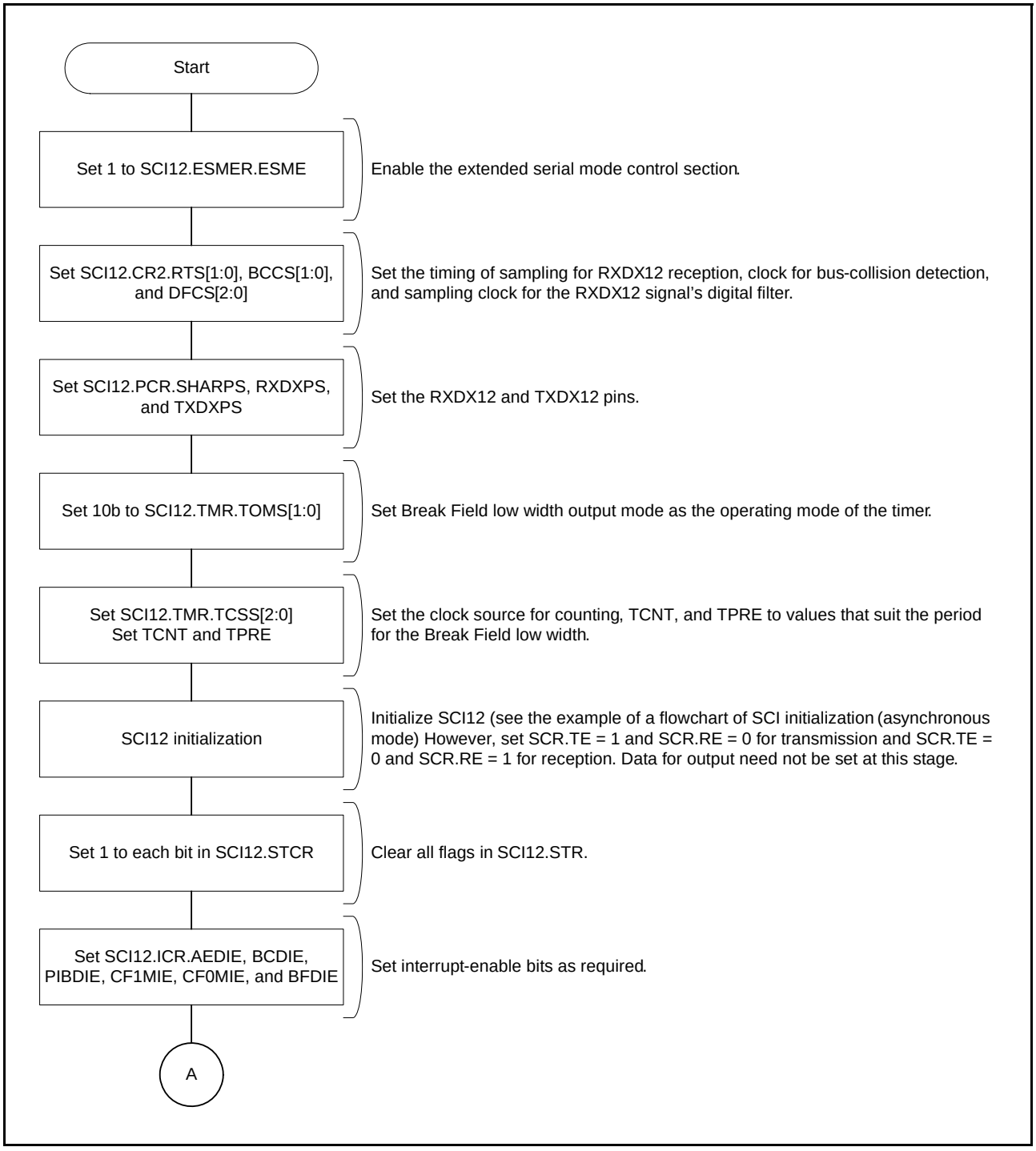



**Figure 26.59 Sample Flowchart for Transmission of a Start Frame (1)**

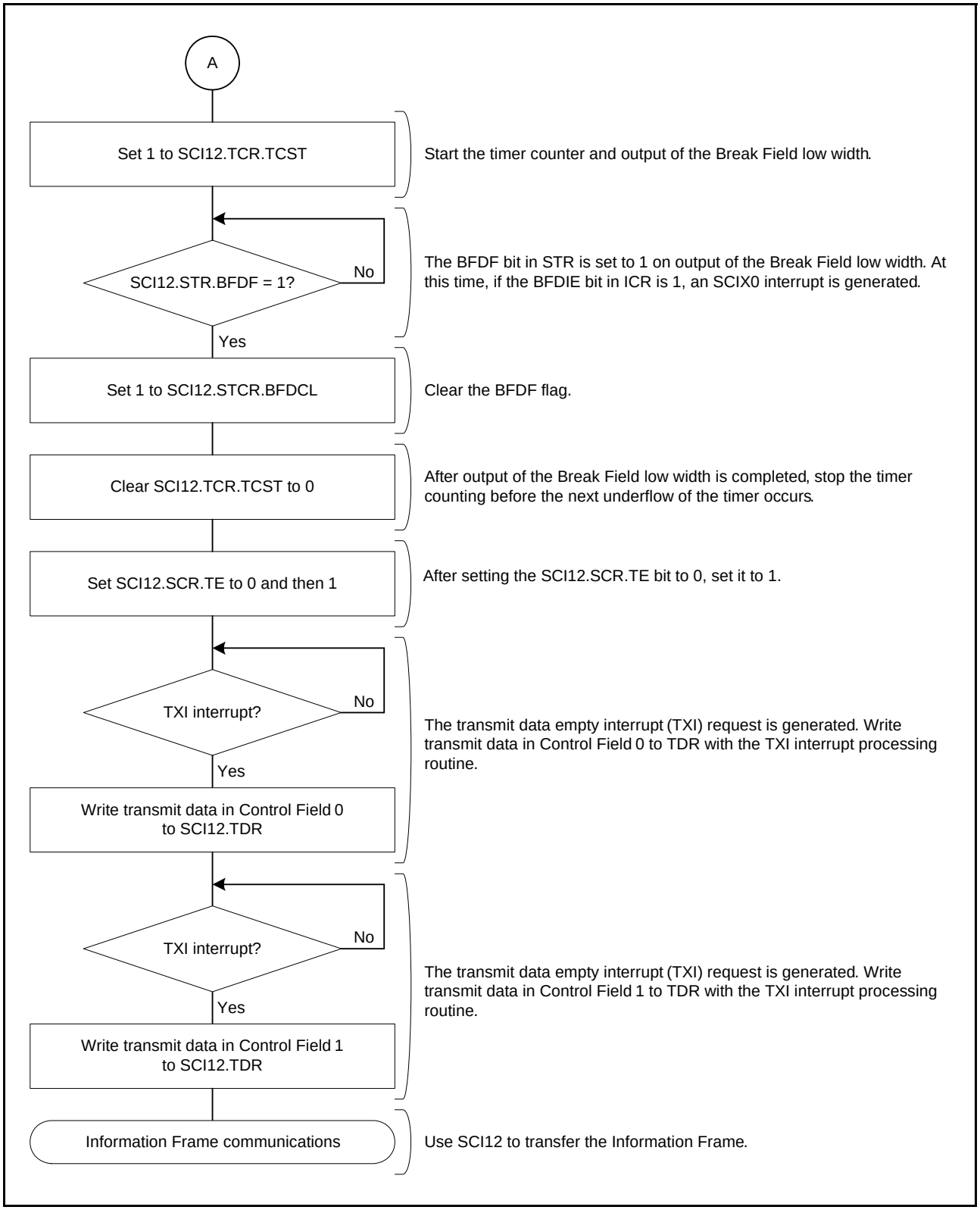



**Figure 26.60 Sample Flowchart for Transmission of a Start Frame (2)**

## 26.9.3 Receiving a Start Frame

The extended serial mode control section is capable of receiving Start Frames with the structures listed in Table 26.27.

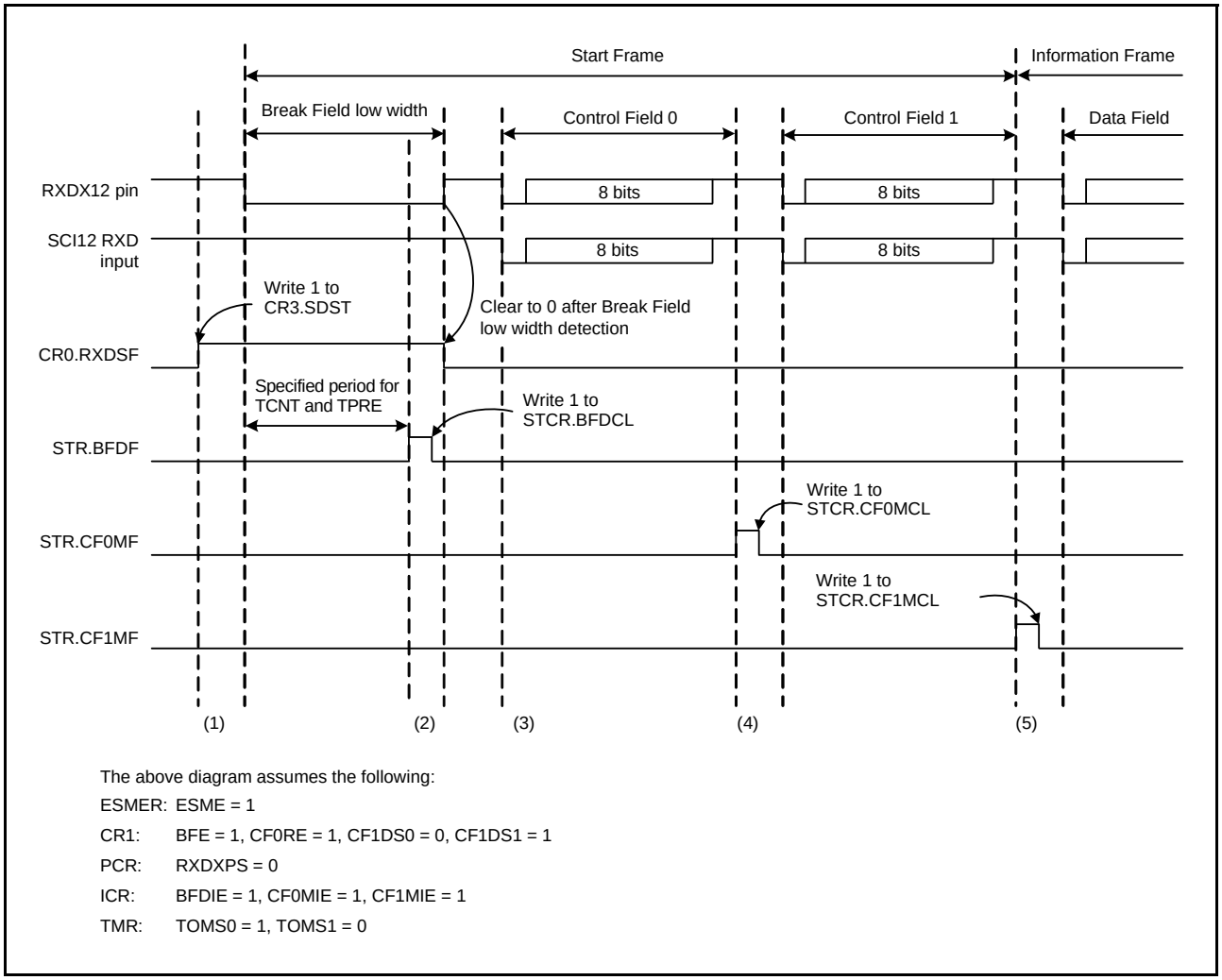
**Table 26.27 Structures of Start Frames**

| Bit Setting | | Structures of Start Frames |
|---|---|---|
| BFE | CF0RE | |
| 0 | 0 | Control Field 1 (8 bits) → Information Frame |
| 0 | 1 | Control Field 0 (8 bits) → Control Field 1 (8 bits) → Information Frame |
| 1 | 0 | Break Field (low width) → Control Field 1 (8 bits) → Information Frame |
| 1 | 1 | Break Field (low width) → Control Field 0 (8 bits) → Control Field 1 (8 bits) → Information Frame |

Figure 26.61 shows an example of operations to receive a Start Frame, which is composed of the Break Field low width, Control Field 0, and Control Field 1. Figure 26.62 and Figure 26.63 are flowcharts for the reception of a Start Frame, and Figure 26.64 is a state transition diagram for the extended serial mode control section.
Operations when the extended serial mode control section is to be used to receive a Start Frame are as listed below. Be sure to use the SCI12 in asynchronous mode.

(1) With Break Field low width detection mode as the operating mode for the timer, writing 1 to the SDST bit in CR3 enables detection of the Break Field low width. RXDX12 input to the SCI12 is disabled at this time.
(2) Low-level input on the RXDX12 pin continuing over a period longer than that corresponding to the settings of TCNT and TPRE is detected as the Break Field low width. At this time, the BFDF bit in STR is set to 1. An SCIX0 interrupt is also generated if the value of the BFDIE bit in ICR is 1.
(3) When the input from the RXDX12 pin goes high after the Break Field low width, the RXDSF bit in CR0 becomes 0 and reception of Control Field 0 by the SCI12 starts.
(4) If the data received in Control Field 0 match the data set in CF0DR, the CF0MF bit in STR is set to 1. An SCIX1 interrupt is also generated if the value of the CF0MIE bit in ICR is 1. Reception of Control Field 1 by the SCI12 starts after that. If the data received in Control Field 0 do not match the data set in CF0DR, a transition to the state prior to Break Field low width detection proceeds.
(5) If the data received in Control Field 1 match the data set in PCF1DR and SCF1DR, the CF1MF bit in STR is set to 1. An SCIX1 interrupt is also generated if the value of the CF1MIE bit in ICR is 1. Transfer of the Information Frame by the SCI12 starts after that. If the data received in Control Field 1 do not match the data set in either or both of PCF1DR and SCF1DR, a transition to the state prior to Break Field low width detection proceeds.

Omit the Break Field and Control Field 0 to suit the structure of the Start Frame.




The above diagram assumes the following:

ESMER: ESME = 1

CR1: BFE = 1, CF0RE = 1, CF1DS0 = 0, CF1DS1 = 1

PCR: RXDXPS = 0

ICR: BFDIE = 1, CF0MIE = 1, CF1MIE = 1

TMR: TOMS0 = 1, TOMS1 = 0

**Figure 26.61 Example of Operations at the Time of Start Frame Reception**

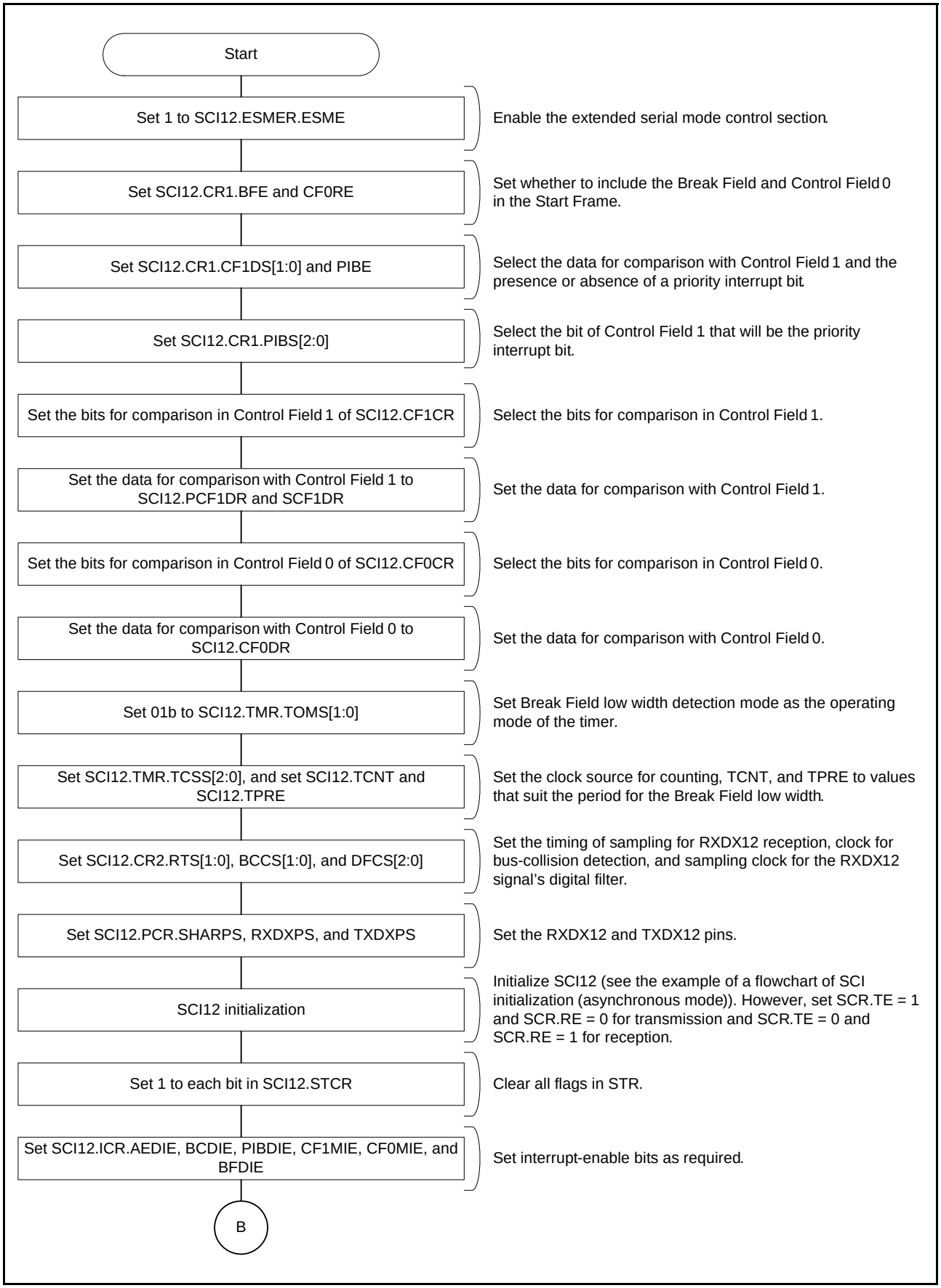



**Figure 26.62 Sample Flowchart for Reception of a Start Frame (1)**

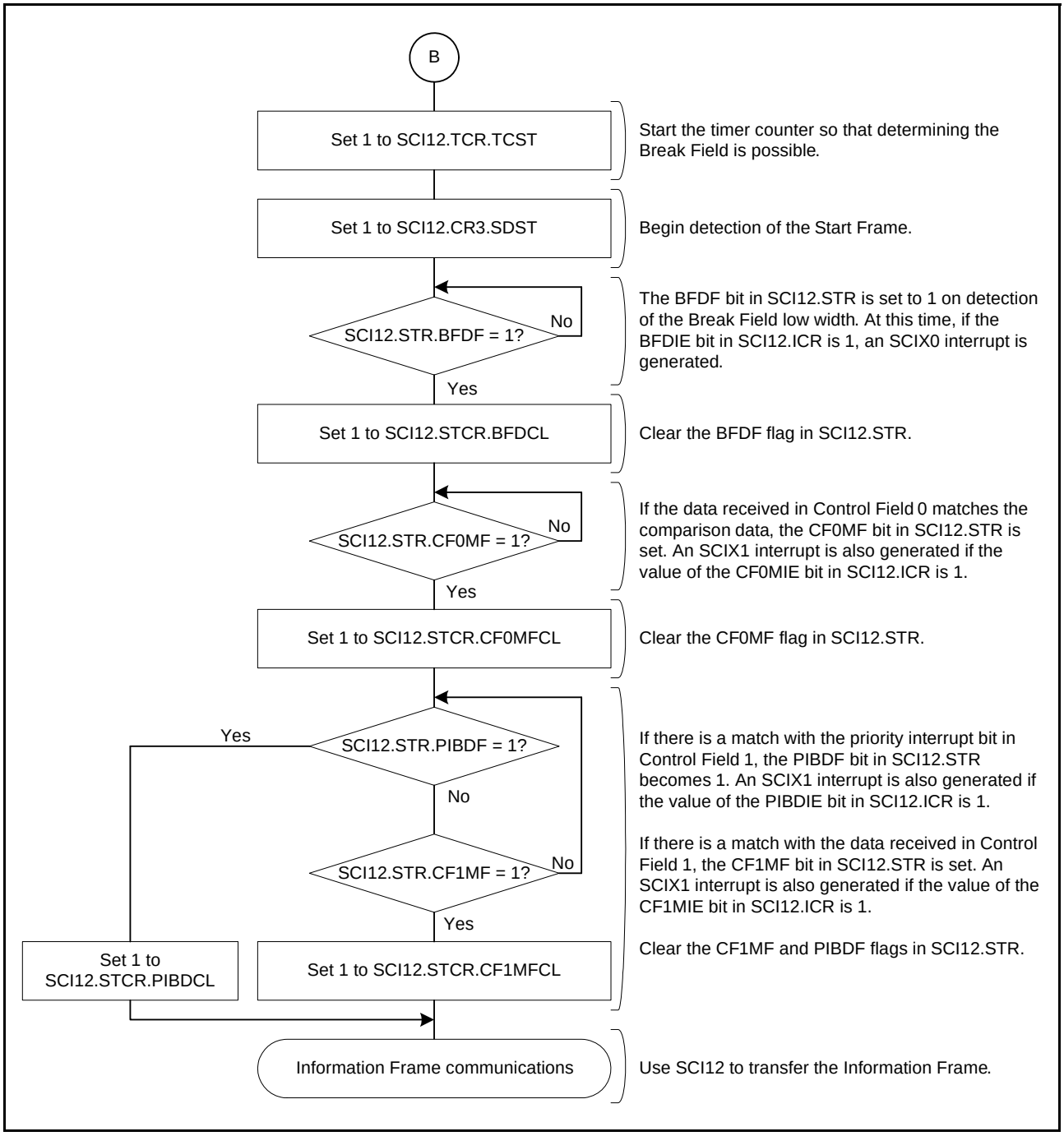



**Figure 26.63 Sample Flowchart for Reception of a Start Frame (2)**

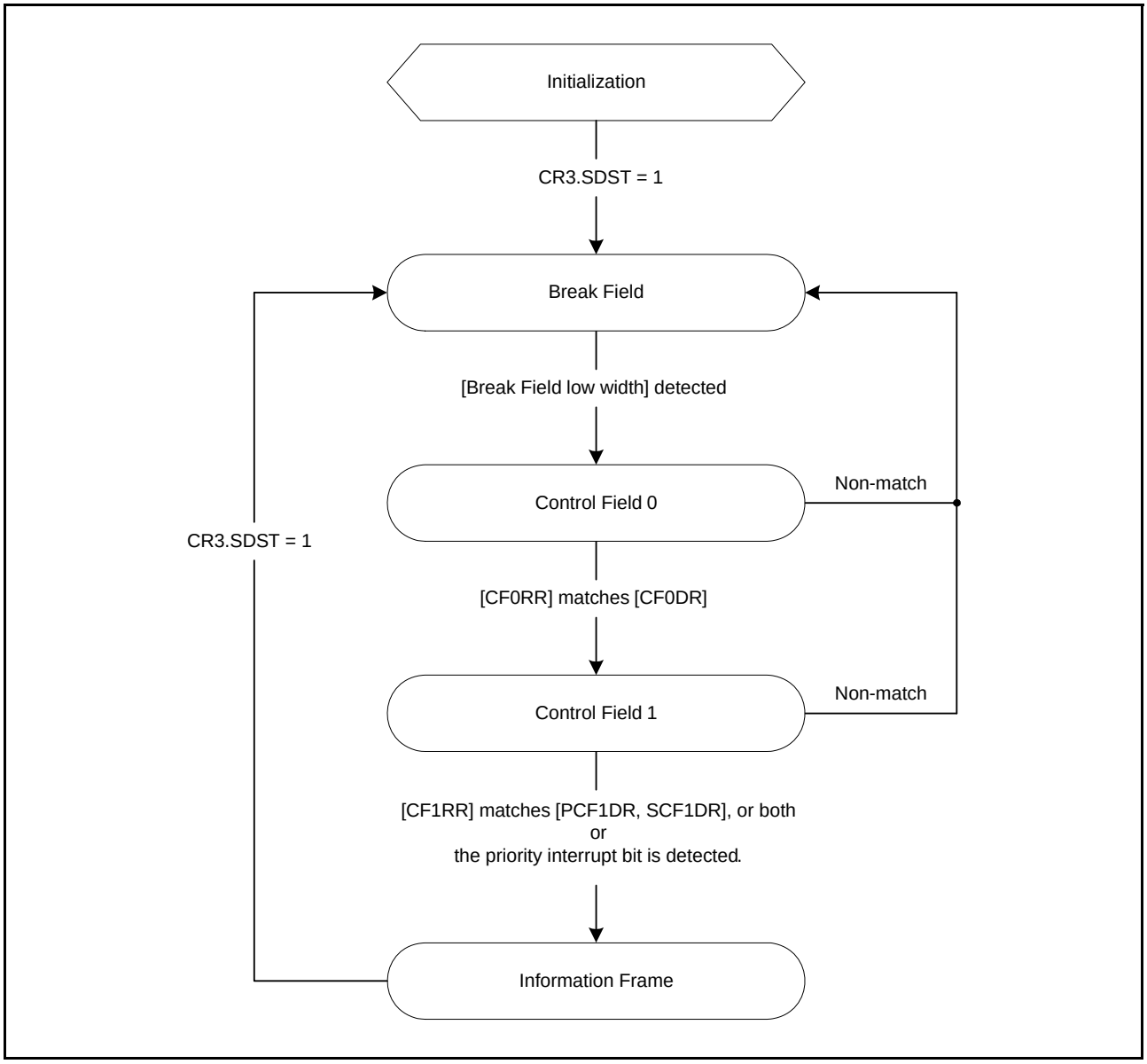


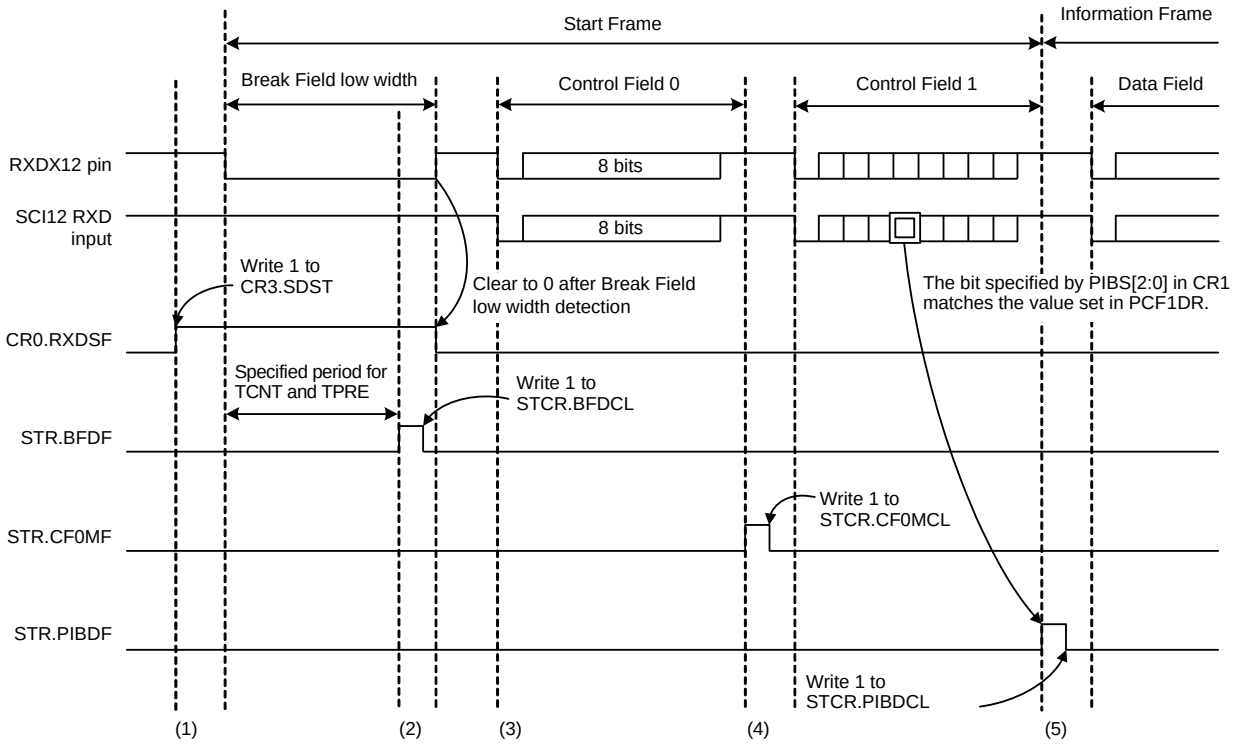

**Figure 26.64 Diagram of State Transitions at the Time of Start Frame Reception**

### 26.9.3.1 Priority Interrupt Bit

Figure 26.65 shows an example of operation in Start Frame reception where a priority interrupt bit is in use. Setting the PIBE bit in CR1 to 1 enables the use of a priority interrupt bit.
Operations of the extended serial mode control section in start Frame reception where a priority interrupt bit is in use are as described below.

Steps (1) to (4) are the same as in Figure 26.61, for Start Frame reception.

(5) If the value of the bit selected by the PIBS[2:0] bits in CR1 matches the corresponding bit in PCF1DR, the PIBDF bit in STR is set to 1. An SCIX1 interrupt is also generated if the value of the PIBDIE bit in ICR is 1. Transfer of the Information Frame by the SCI12 starts after that. If the data received in Control Field 1 do not match the data set in either or both of PCF1DR and SCF1DR and the priority interrupt bit is not detected, a transition to the state prior to Break Field low width detection proceeds.




The above diagram assumes the following:

ESMER: ESME = 1
CR1: BFE = 1, PIBE = 1, CF0RE = 0, CF1DS[1:0] = 10b, PIBS[2:0] = 011b
PCR: RXDXPS = 0
ICR: BFDIE = 1, CF0MIE = 1, PIBDIE = 1
TMR: TOMS0 = 1, TOMS1 = 0

**Figure 26.65 Example of Operations at the Time of Start Frame Reception (with Priority Interrupt Bit in Use)**

### 26.9.4 Detection of Bus Collisions

Detection of bus collisions operate for cases where output of the Break Field low width and transmission of data by the SCI12 are in progress when the ESMER.ESME and the SCI12.SCI.TE are set to 1.

Figure 26.66 shows an example of operations with bus collision detection. Signals output through TXDX12 and input through RXDX12 are sampled with the bus-collision detection clock set with CR2.BCCS[1:0] as the sampling clock, and the BCDF bit in STR is set to 1 if the signals fail to match three times in a row. An SCIX2 interrupt is also generated if the value of the BCDIE bit in ICR is 1.

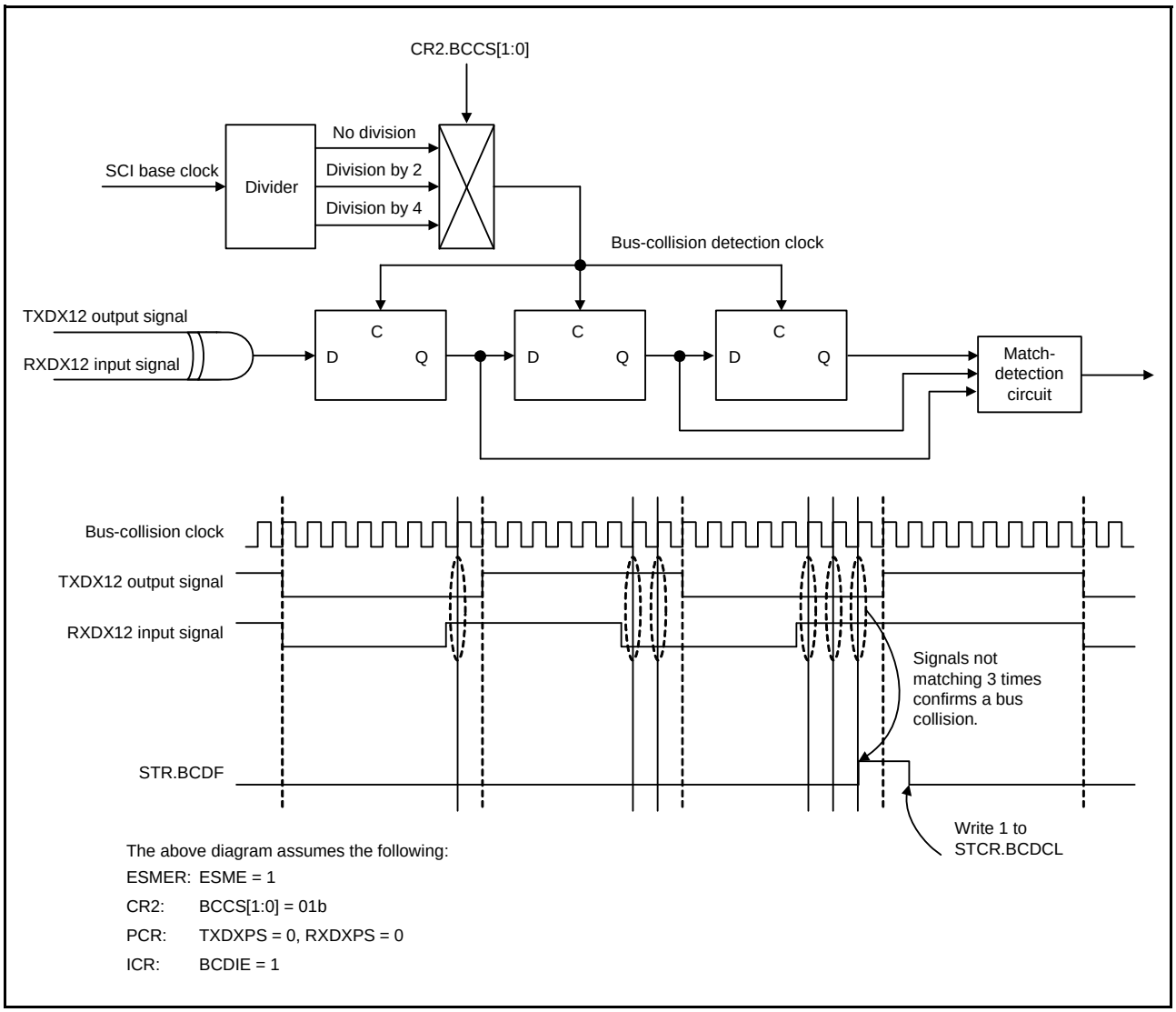



**Figure 26.66 Example of Operations with Bus-Collision Detection**

### 26.9.5 Digital Filter for Input on the RXDX12 Pin

Signals input through the RXDX12 pin can be passed through a digital filter before they are conveyed to the internal circuits. The digital filter consists of three flip-flop circuit stages connected in series and a match-detecting circuit. The DFCS[2:0] bits in CR2 select the sampling clock for the RXDX12 pin input signals. If the outputs of all three latches match, the given level is conveyed to subsequent circuits. If the levels do not match, the previous value is retained. In other words, levels are confirmed as being the signal if they are retained for at least three cycles of the sampling clock but judged to be noise rather than changes in the signal level if they change within three cycles of the sampling clock. Figure 26.67 shows an example of operations with the digital filter.

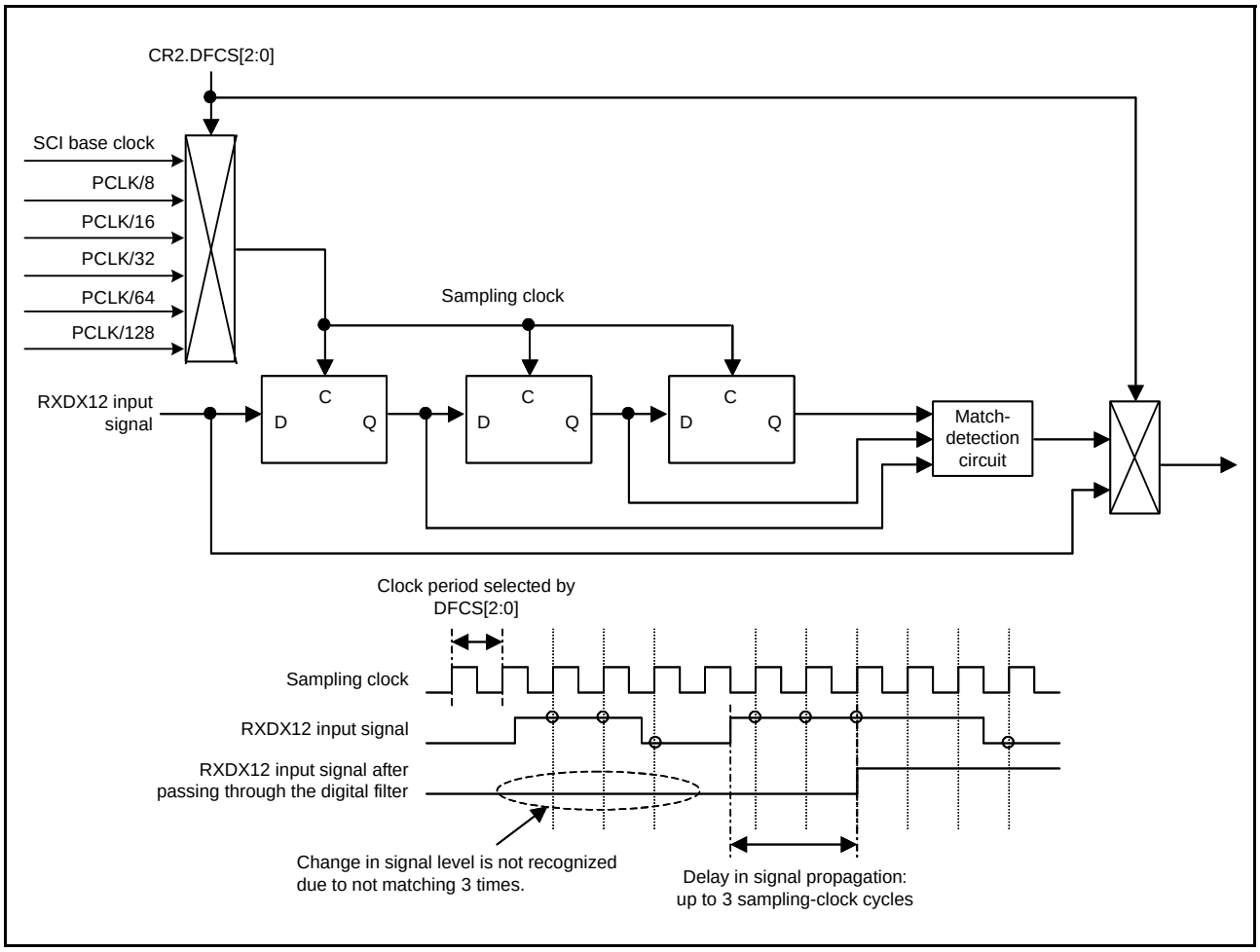
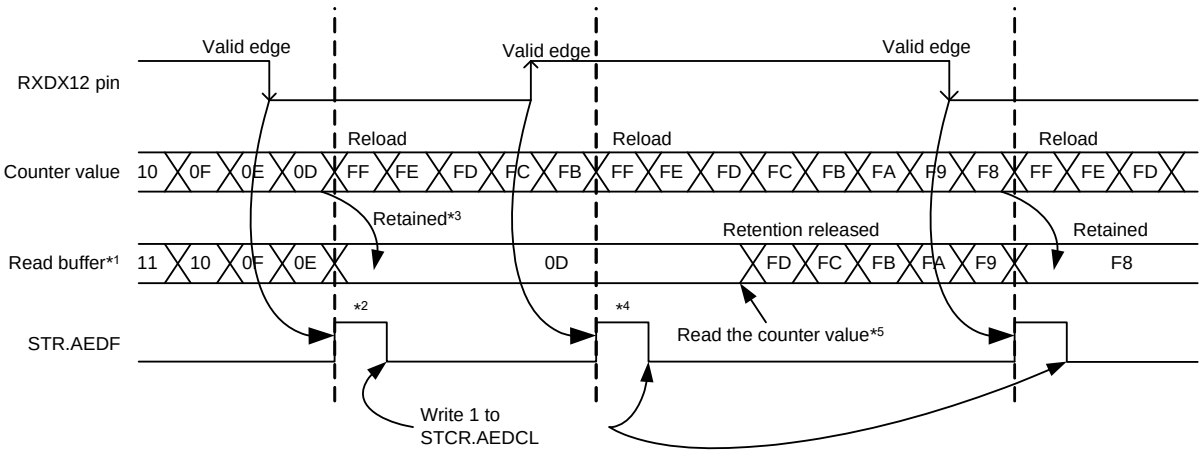



**Figure 26.67 Example of Operations with the Digital Filter**

### 26.9.6 Bit-Rate Measurement

The bit-rate measurement function measures the intervals between rising and falling edges and between falling and rising edges of the signal input from the RXDX12 pin. Figure 26.68 shows an example of operations for bit-rate measurement.

(1) Writing 1 to the BRME bit in CR0 enables bit-rate measurement. Only set BRME to 1 when you wish to proceed with bit-rate measurement. Furthermore, bit-rate measurement will not proceed during a Break Field, even if BRME is set to 1.
(2) After detection of the Break Field low width, bit-rate measurement starts when the level input on the RXDX12 pin becomes high.
(3) Once bit-rate measurement has started, counter values from the timer are retained in the read buffers on the input of valid edges from the RXDX12 pin (rising and falling edges) and the counter is reloaded. An SCIX3 interrupt is also generated if the value of the AEDIE bit in ICR is 1. Retention by TCNT and TPRE is released by reading these registers.
(4) The bit rate as calculated from the values counted during intervals between valid edges can be used for adjusting the rate by changing the settings of the SCI12. To disable the bit-rate measurement after a match with Control Field 1, write 0 to the BRME bit in CR0.




Note 1. The contents of the read buffers are read by reading TCNT and TPRE.
Note 2. The AEDF bit in STR becomes 1 on detection of a valid edge.
Note 3. The current counter value is retained in the read buffers on detection of a valid edge.
Note 4. The value retained in the read buffers due to one valid edge is not updated to the new value even if a further valid edge is detected.
Note 5. Retention of a value in a read buffer due to a valid edge is released by reading TCNT or TPRE.

**Figure 26.68 Example of Operations for Bit-Rate Measurement**

### 26.9.7 Selectable Timing for Sampling Data Received through RXDX12

The extended serial mode control section provides a way of adjusting the timing for the sampling of data received through the RXDX12 pin of an SCI12 by setting the RTS0 and RTS1 bits in CR2 to select the rising edges of 8, 10, 12, or 14 cycles of the SCI base clock. If the value of the ABCS bit in SEMR is 1, the bits select the rising edges of 4, 5, 6, or 7 cycles of the PCLK clock of the SCI12. Figure 26.69 shows timing for the sampling of data received through RXDX12.

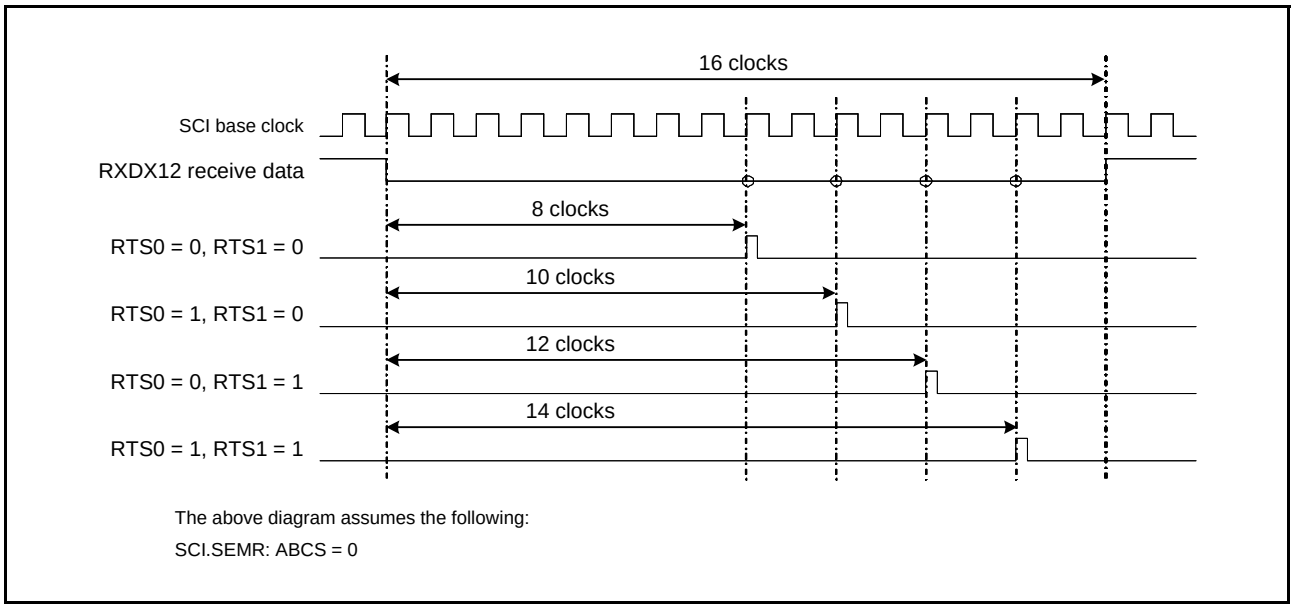



**Figure 26.69 Timing for Sampling of Data Received through RXDX12**

## 26.9.8 Timer

The timer has the following operating modes.

### (1) Break Field Low Width Output Mode

This mode is for output through the TXDX12 pin of the low level over the Break Field low width at the time of transmitting a Start Frame. Setting TOMS0 to 0 and TOMS1 to 1 in TMR switches operation to Break Field low width output mode. The TCSS[2:0] bits in TMR select the clock source for the counter. When the TCST bit in TCR is set to 1, the output on the TXDX12 pin goes to the low level and counting starts. When the timer underflows, the output on the TXDX12 pin goes to the high level and the BFDF bit in the STR is set to 1. An SCIX0 interrupt is also generated if the value of the BFDIE bit in ICR is 1. When 0 is written to the TCST bit in TCR, counting stops after reloading of TPRE and TCNT. After output of the Break Field low width is completed, stop the timer before it underflows again. Figure 26.70 shows an example of operations in Break Field low width output mode.

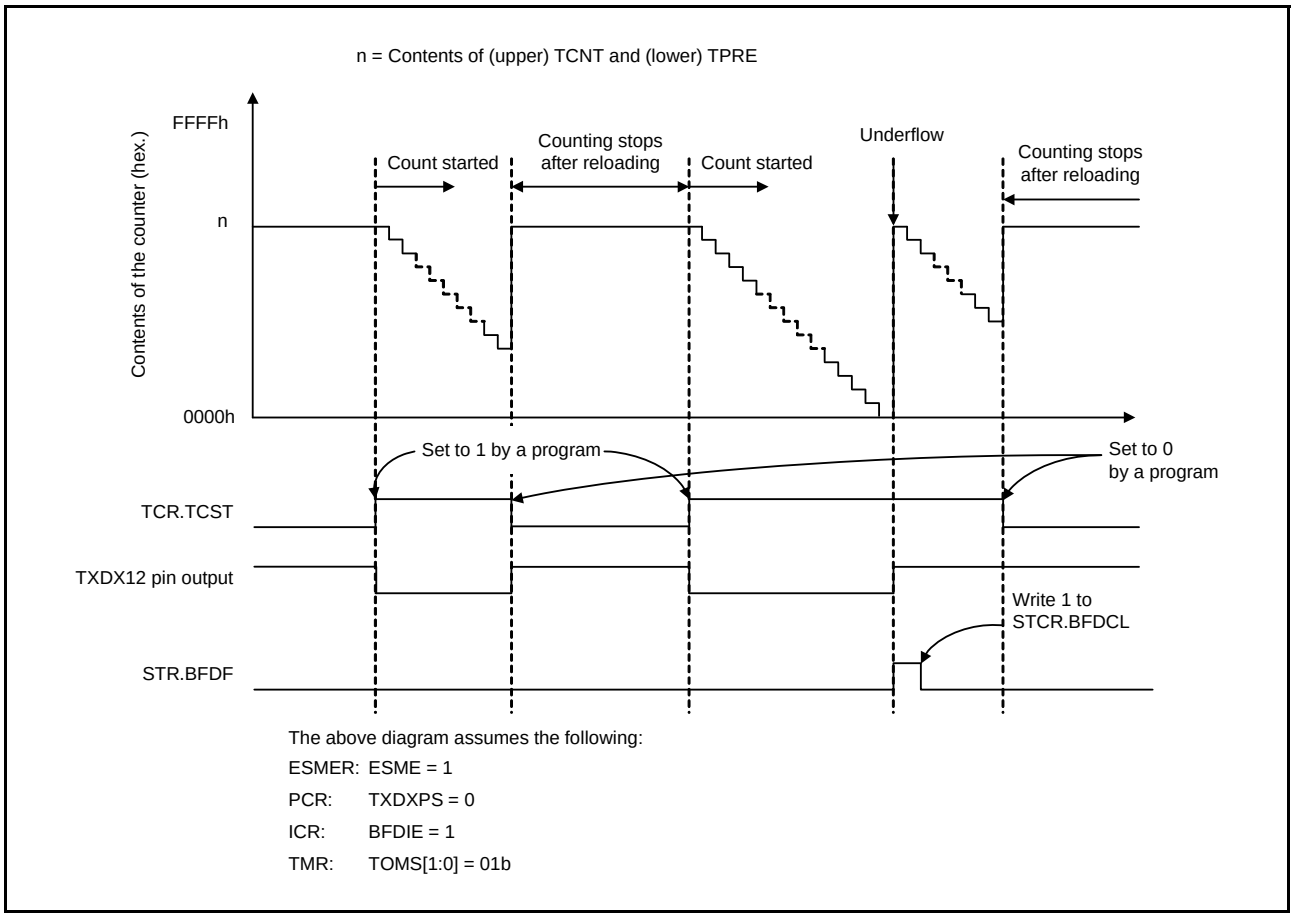



**Figure 26.70 Example of Operations in Break Field Low Width Output Mode**

### (2) Break Field Low Width Determination Mode

This mode is for determining the Break Field low width in the input signal on the RXDX12 pin at the time of receiving a Start Frame. Setting TOMS0 to 1 and TOMS1 to 0 in TMR switches operation to Break Field low width determination mode. The TCSS[2:0] bits in TMR select the clock source for the counter. When the TCST bit in TCR is set to 1, the interface enters the Break Field low width determinable state. Determination starts when a low level is input from the RXDX12 pin. When a high level is then input on the RXDX12 pin, TPRE and TCNT are reloaded and the interface enters the Break Field low width determinable state. When the timer underflows during Break Field low width determination, the BFDF bit in STR is set to 1. An SCIX0 interrupt is also generated if the value of the BFDIE bit in ICR is 1. If an underflow of the timer during data transfer cause a problem in the form of interrupt generation, stop the timer after Break Field low width determination. Figure 26.71 shows an example of operations in Break Field low width output mode.

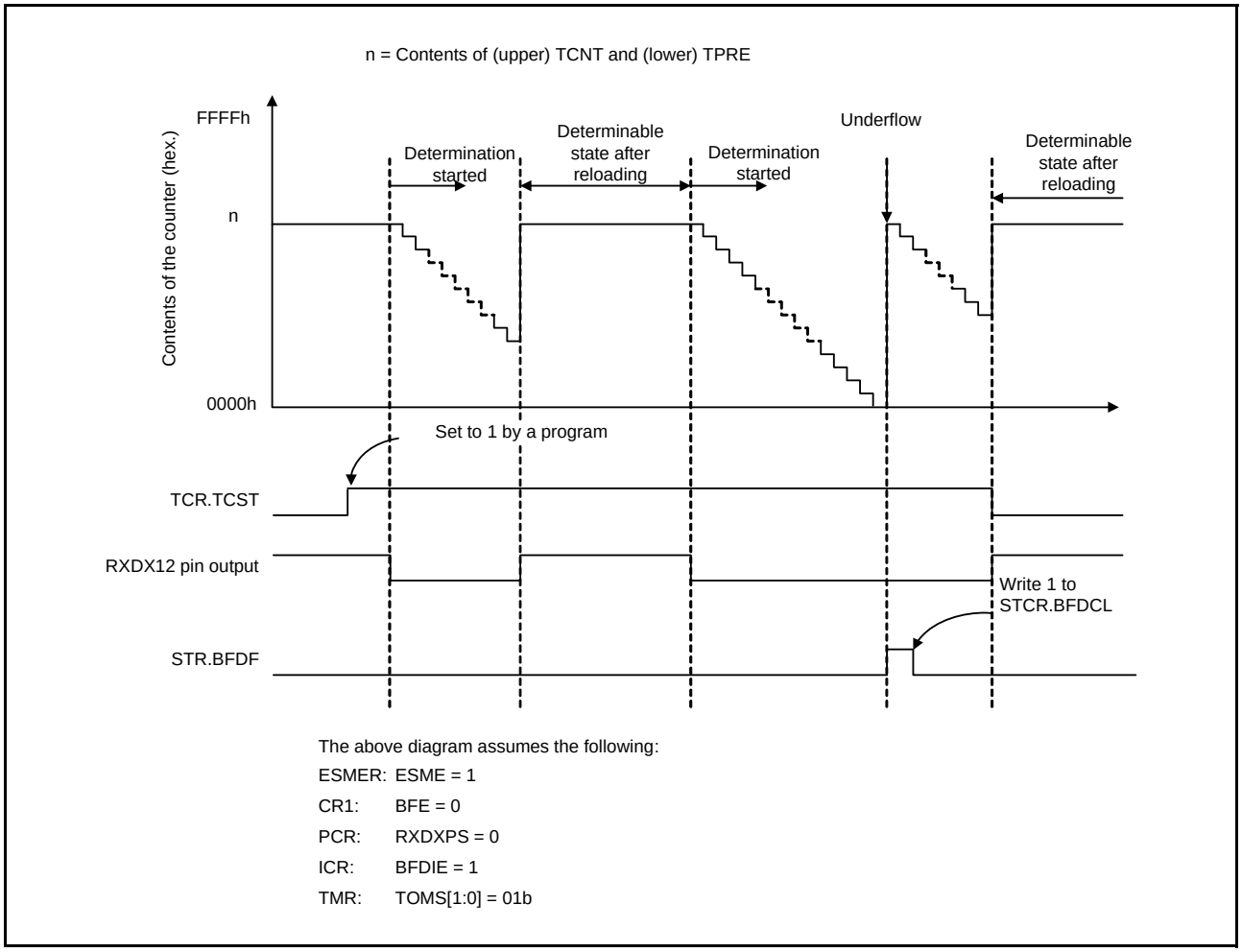



**Figure 26.71 Example of Operations in Break Field Low Width Determination Mode**

### (3) Timer Mode

This mode is for counting cycles of the internal clock as the clock source. Setting TOMS0 to 0 and TOMS1 to 0 in TMR switches operation to timer mode. The TCSS[2:0] bits in TMR select the clock source for the counter. Counting starts when 1 is written to the TCST bit in TCR and stops when 0 is written to TCST. TPRE and TCNT both count down. TPRE counts cycles of the clock source for counting, and underflows of TPRE provide the clock source for counting by TCNT. When the timer underflows, the BFDF bit in STR is set to 1. An SCIX0 interrupt is also generated if the value of the BFDIE bit in ICR is 1.

## 26.10 Noise Cancellation Function

Figure 26.72 shows the configuration of the noise filter used for noise cancellation. The noise filter consists of two stages of flip-flop circuits and a match-detection circuit. When the level on the pin matches in three consecutive samples taken at the set sampling interval, the matching level continues to be conveyed internally until the level on the pin again matches in three consecutive samples.

In asynchronous mode, the noise cancellation function can be applied on the RXDn input signal. The period of the base clock (1/16th of a bit-period when SEMR.ABCS = 0 and 1/8th of a bit-period when SEMR.ABCS = 1) is the sampling interval.

In simple I2C mode, the noise cancellation function can be applied on the SSDAn and SSCLn input signals. The sampling clock is the clock signal produced by frequency-dividing the signal from the clock source for the internal baud-rate generator by one, two, four, or eight as selected by the setting of the SNFR.NFCS[2:0] bits.

If the base clock is stopped with the noise filter enabled and then the clock input is started again, the noise filter operation resumes from where the clock was stopped. If SCR.TE and SCR.RE are set to 0 during base clock input, all of the noise filter flip-flop values are initialized to 1. Accordingly, if the input data is 1 when reception operation resumes, it is determined that a level match is detected and is conveyed to the internal signal. When the level being input corresponds to 0, the initial output of the noise filter is retained until the level matches in three consecutive samples.

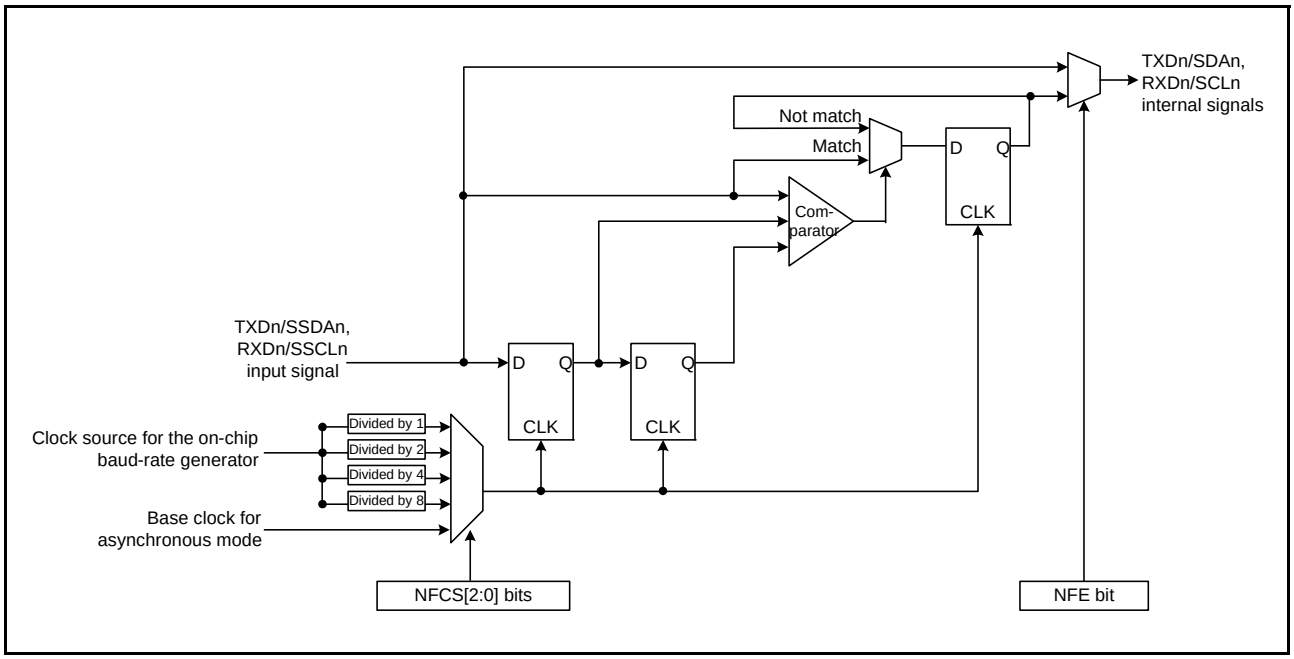



**Figure 26.72 Block Diagram of Digital Noise Filter Circuit**

## 26.11 Interrupt Sources

### 26.11.1 Buffer Operations for TXI and RXI Interrupts

If the conditions for a TXI and RXI interrupt are satisfied while the interrupt status flag in the interrupt controller is 1, the interrupt controller does not output the interrupt request but retains it internally (with a capacity for retention of one request per source).
When the value of the interrupt status flag in the interrupt controller becomes 0, the interrupt request retained within the interrupt controller is output. The internally retained interrupt request is automatically discarded once the actual interrupt is output. Clearing of the corresponding interrupt enable bit (the TIE or RIE bit in the SCR) can also be used to discard an internally retained interrupt request.

### 26.11.2 Interrupts in Asynchronous Mode, Clock Synchronous Mode, and Simple SPI Mode

Table 26.28 lists interrupt sources in asynchronous mode, clock synchronous mode, and simple SPI mode. A different interrupt vector is assigned to each interrupt source, and individual interrupt sources can be enabled or disabled with the enable bits in SCR.
If the SCR.TIE bit is 1, a TXI interrupt request is generated when transmit data are transferred from the TDR to the TSR. A TXI interrupt request can also be generated by setting the SCR.TE bit to 1 after setting the SCR.TIE bit to 1 or by using a single instruction to set the SCR.TE and SCR.TIE bit to 1 at the same time. A TXI interrupt request can activate the DTC to handle data transfer.
A TXI interrupt request is not generated by setting the SCR.TE bit to 1 while the setting of the SCR.TIE bit is 0 or by setting the SCR.TIE bit to 1 while the setting of the SCR.TE bit is 1.*1
When new data are not written by the time of transmission of the last bit of the current transmit data and the setting of the SCR.TEIE bit is 1, the SSR.TEND flag becomes 1 and a TEI interrupt request is generated. Furthermore, when the setting of the SCR.TE bit is 1, the SSR.TEND flag retains the value 1 until further transmit data are written to the TDR, and setting the SCR.TEIE bit to 1 leads to the generation of a TEI interrupt request.
Writing data to the TDR leads to clearing of the SSR.TEND flag and, after a certain time, discarding of the TEI interrupt request.
If the SCR.RIE bit is 1, an RXI interrupt request is generated when received data are stored in the RDR. An RXI interrupt request can activate the DTC to handle data transfer.
Setting of any from among the ORER, FER, and PER flags in the SSR to 1 while the SCR.RIE bit is 1 leads to the generation of an ERI interrupt request. An RXI interrupt request is not generated at this time. Clearing all three flags (ORER, FER, and PER) leads to discarding of the ERI interrupt request.

Note 1. To temporarily prohibit TXI interrupts at the time of transmission of the last of the data and so on when you wish a new round of transmission to start after handling of the transmission-completed interrupt, control prohibiting and permitting of the interrupt by using the interrupt request enable bit in the interrupt controller rather than using the SCR.TIE bit. This can prevent the suppression of TXI interrupt requests in the transfer of new data.

**Table 26.28 Interrupt Sources**

| Name | **Interrupt Source** | Interrupt Flag | DTC Activation | Priority |
|---|---|---|---|---|
| ERI | Receive error | ORER, FER, or PER | Not possible | High |
| RXI | Receive data full | — | Possible | ↑ |
| TXI | Transmit data empty | — | Possible | |
| TEI | Transmit end | TEND | Not possible | Low |

### 26.11.3 Interrupts in Smart Card Interface Mode

Table 26.29 lists interrupt sources in smart card interface mode. A transmit end interrupt (TEI) request cannot be used in this mode.

**Table 26.29 SCI Interrupt Sources**

| Name | Interrupt Source | Interrupt Flag | DTC Activation | Priority |
|---|---|---|---|---|
| ERI | Receive error or error signal detection | ORER, PER, or ERS | Not possible | High |
| RXI | Receive data full | — | Possible | ↑ |
| TXI | Transmit data empty | TEND | Possible | Low |

Data transmission/reception using the DTC is also possible in smart card interface mode, similar to in the normal SCI mode. In transmission, when the TEND flag in SSR is set to 1, a TXI interrupt request is generated. This TXI interrupt request activates the DTC allowing transfer of transmit data if the TXI request is specified beforehand as a source of DTC activation. The TEND flag is automatically cleared to 0 when the DTC transfers the data.

If an error occurs, the SCI automatically retransmits the same data. During the retransmission, the TEND flag is kept to 0 and the DTC is not activated. Therefore, the SCI and DTC automatically transmit the specified number of bytes, including retransmission in the case of error occurrence. However, the ERS flag in SSR is not automatically cleared to 0 at error occurrence. Therefore, the ERS flag must be cleared by previously setting the RIE bit in SCR to 1 to enable an ERI interrupt request to be generated at error occurrence.

When transmitting/receiving data using the DTC, be sure to make settings to enable the DTC before making SCI settings. For DTC settings, refer to section 16, Data Transfer Controller (DTCa).

In reception, an RXI interrupt request is generated when receive data is set to RDR. This RXI interrupt request activates the DTC allowing transfer of receive data if the RXI request is specified beforehand as a source of DTC activation. If an error occurs, the error flag is set. Therefore, the DTC is not activated and an ERI interrupt request is issued to the CPU instead; the error flag must be cleared.

### 26.11.4 Interrupts in Simple I2C Mode

The interrupt sources in simple I2C mode are listed in Table 26.30. The STI interrupt is allocated to the transmit end interrupt (TEI) request. The receive error interrupt (ERI) request cannot be used.

The DTC can also be used to handle transfer in simple I2C mode.

When the value of the IICINTM bit in SIMR2 is 1, an RXI request will be generated on the falling edge of the SSCLn signal for the eighth bit. If the RXI has been set up as an activating request for the DTC beforehand, the RXI request will activate the DTC to handle transfer of the received data. Furthermore, a TXI request is generated on the falling edge of the SSCLn signal for the ninth bit (acknowledge bit). If the TXI has been set up as an activating request for the DTC beforehand, the TXI request will activate the DTC to handle transfer of the transmit data.

When the value of the IICINTM bit in SIMR2 is 0, an RXI request (ACK detection) if the input on the SSDAn pin is at the low level or a TXI request (NACK detection) if the input on the SSDAn pin is at the high level will be generated on the rising edge of the SSCLn signal for the ninth bit (acknowledge bit). If the RXI has been set up as an activating request for the DTC beforehand, the RXI request will activate the DTC to handle transfer of the received data.

Also, if the DTC is used for data transfer in reception or transmission, be sure to set up and enable the DTC before setting up the SCI.

When the IICSTAREQ, IICRSTAREQ, and IICSTPREQ bits in SIMR3 are used to generate a start condition, restart condition, or stop condition, the STI request is issued when generation is complete.

**Table 26.30 SCI Interrupt Sources**

| Name | **Interrupt Source** | Interrupt Flag | DTC Activation | Priority |
|---|---|---|---|---|
| RXI | Reception, ACK detection | — | Possible | High |
| TXI | Transmission, NACK detection | — | Possible*1 | ↑ |
| STI | Completion of generating a start, restart, or stop condition | IICSTIF | Not possible | Low |

Note 1. Activation of the DTC is only possible when the SIMR2.IICINTM bit is 1 (reception and transmission interrupts selected).

### 26.11.5 Interrupts from the Extended Serial Mode Control Section

The extended serial mode control section has a total of six types of interrupt request for generating the SCIX0 interrupt (Break Field low width detected), SCIX1 interrupt (Control Field 0 match, Control Field 1 match, priority interrupt bit detected), SCIX2 interrupt (bus collision detected), and SCIX3 interrupt (valid edge detected). When any of the interrupt factors is generated, the corresponding status flag is set to 1. Details of all of the interrupt requests are listed in Table 26.31.

**Table 26.31 Interrupt Sources of the Extended Serial Mode Control Section**

| Interrupt Request | Status Flag | Interrupt Factors |
|---|---|---|
| SCIX0 interrupt (Break Field low width detected) | BFDF | • Detection of a Break Field low width longer than the interval corresponding to the timer setting<br>• Completion of the output of a Break Field low width over the interval corresponding to the timer setting<br>• Underflow of the timer |
| SCIX1 interrupt (Control Field 0 match) | CF0MF | The data received in Control Field 0 matching the value set in CF0DR |
| SCIX1 interrupt (Control Field 1 match) | CF1MF | The data received in Control Field 1 matching the value set in PCF1DR or SCF1DR |
| SCIX1 interrupt (priority interrupt bit detected) | PIBDF | The value of the bit specified as the priority interrupt bit matching the value set in PCF1DR |
| SCIX2 interrupt (bus collision detected) | BCDF | The output level on the TXDX12 pin and the input level on the RXDX12 pin not matching on three consecutive cycles of the bus-collision detection clock |
| SCIX3 interrupt (valid edge detected) | AEDF | Detection of a valid edge during bit-rate measurement |

## 26.12 Event Linking

By employing interrupt request signals as event signals, SCI5 is able to provide linked operation through the event link controller (ELC) for modules selected in advance.
Event signals can be output regardless of the values of the corresponding interrupt request enable bits.

(1) Error (reception error, error signal detected) event output

- Indicates abnormal termination due to a parity error during reception in asynchronous mode.
- Indicates abnormal termination due to a framing error during reception in asynchronous mode.
- Indicates abnormal termination due to an overrun error during reception.
- Indicates detection of the error signal during transmission in smart card interface mode.

(2) Receive data full event output

- Indicates that received data have been set in the receive data register (RDR).
- Indicates that ACK has been detected if the SIMR2.IICINTM bit is 0 in simple $I^2C$ mode.
- Indicates that the eighth-bit SSCL5 falling edge has been detected if the SIMR2.IICINTM bit is 1 in simple $I^2C$ mode.
- When the SIMR2.IICINTM bit is 1 during master transmission in simple $I^2C$ mode, set the event link controller (ELC) so that receive data full events are not used.

(3) Transmit data empty event output

- Indicates that the SCR.TE bit has been changed from 0 to 1.
- Indicates that transmit data have been transferred from the transmit data register (TDR) to the transmit shift register (TSR).
- Indicates that transmission has been completed in smart card interface mode.
- Indicates that NACK has been detected if the SIMR2.IICINTM bit is 0 in simple $I^2C$ mode.
- Indicates that the ninth-bit SSCL5 falling edge has been detected if the SIMR2.IICINTM bit is 1 in simple $I^2C$ mode.

(4) Transmit end event output

- Indicates the completion of transmission.
- Indicates that the starting condition, resumption condition, or termination condition has been generated in simple $I^2C$ mode.

## 26.13 Usage Notes

### 26.13.1 Setting the Module Stop Function

Module stop control register B (MSTPCRB) is used to stop and start SCI operations. With the value after a reset, SCI operations are stopped. The registers of the modules only become accessible after release from the module stop state. For details, refer to section 11, Low Power Consumption.

### 26.13.2 Break Detection and Processing

When a framing error is detected, a break can be detected by reading the RXDn pin value directly. In a break, the input from the RXDn pin becomes all 0s, and so the FER flag in SSR is set to 1 (framing error), and the PER flag in SSR may also be set to 1 (parity error). The SCI continues the receive operation even after a break is received. Therefore, note that even if the FER flag is cleared to 0 (no framing error), it will be set to 1 again. When the SEMR.RXDESEL bit is 1, the SCI sets the SSR.FER flag to 1 and stops receiving operation until a start bit of the next data frame is detected. If the SSR.FER flag is cleared to 0 at this time, the SSR.FER flag retains 0 during the break. When the RXDn pin is set to 1 and the break ends, detecting the beginning of the start bit at the first falling edge of the RXDn pin allows the SCI to start the receiving operation.

### 26.13.3 Mark State and Production of Breaks

When the SCR.TE bit is 0 (serial transmission is disabled), setting the I/O port function makes selection of the level and direction (input or output) of the TXDn pin possible. If this is done, the TXDn pin can be placed in the mark state to send a break at the time of data transmission. Until the SCR.TE bit is set to 1 (serial transmission is enabled), the I/O port function is used to set the TXDn pin to output 1 and set the pin mode to a general I/O port pin, and thus place the transfer circuit in the mark state (state of having the value 1). On the other hand, to output a break at the time of data transmission, set the TXDn pin to output 0 and make the pin mode settings for a general I/O port pin. When the SCR.TE bit is set to 0, the transmitter is initialized regardless of the current state of transmission.

### 26.13.4 Receive Error Flags and Transmit Operations (Clock Synchronous Mode and Simple SPI Mode)

Transmission cannot be started when a receive error flag (ORER) in SSR is set to 1, even if data is written to TDR. Be sure to clear the receive error flags to 0 before starting transmission. Note also that the receive error flags cannot be cleared to 0 even if the RE bit in SCR is cleared to 0 (serial reception is disabled).

### 26.13.5 Writing Data to TDR

Data can always be written to TDR. However, if new data is written to TDR when transmit data is remaining in TDR, the previous data in TDR is lost because it has not been transferred to TSR yet. Be sure to write transmit data to TDR in the TXI interrupt request processing routine.

### 26.13.6 Restrictions on Clock Synchronous Transmission (Clock Synchronous Mode and Simple SPI Mode)

When the external clock source is used as a synchronization clock, the following restrictions apply.

#### (1) Start of transmission

Update TDR by the CPU or DTC and wait for at least five PCLK cycles before allowing the transmit clock to be input (see Figure 26.73).

#### (2) Continuous transmission

(a) Write the next transmit data to TDR or TDRL before the falling edge of the transmit clock (bit 7) (see Figure 26.73).

(b) When updating TDR after bit 7 has started to transmit, update TDR while the synchronization clock is in the low-level period, and set the high-level width of the transmit clock (bit 7) to four PCLK cycles or longer (see Figure 26.73).

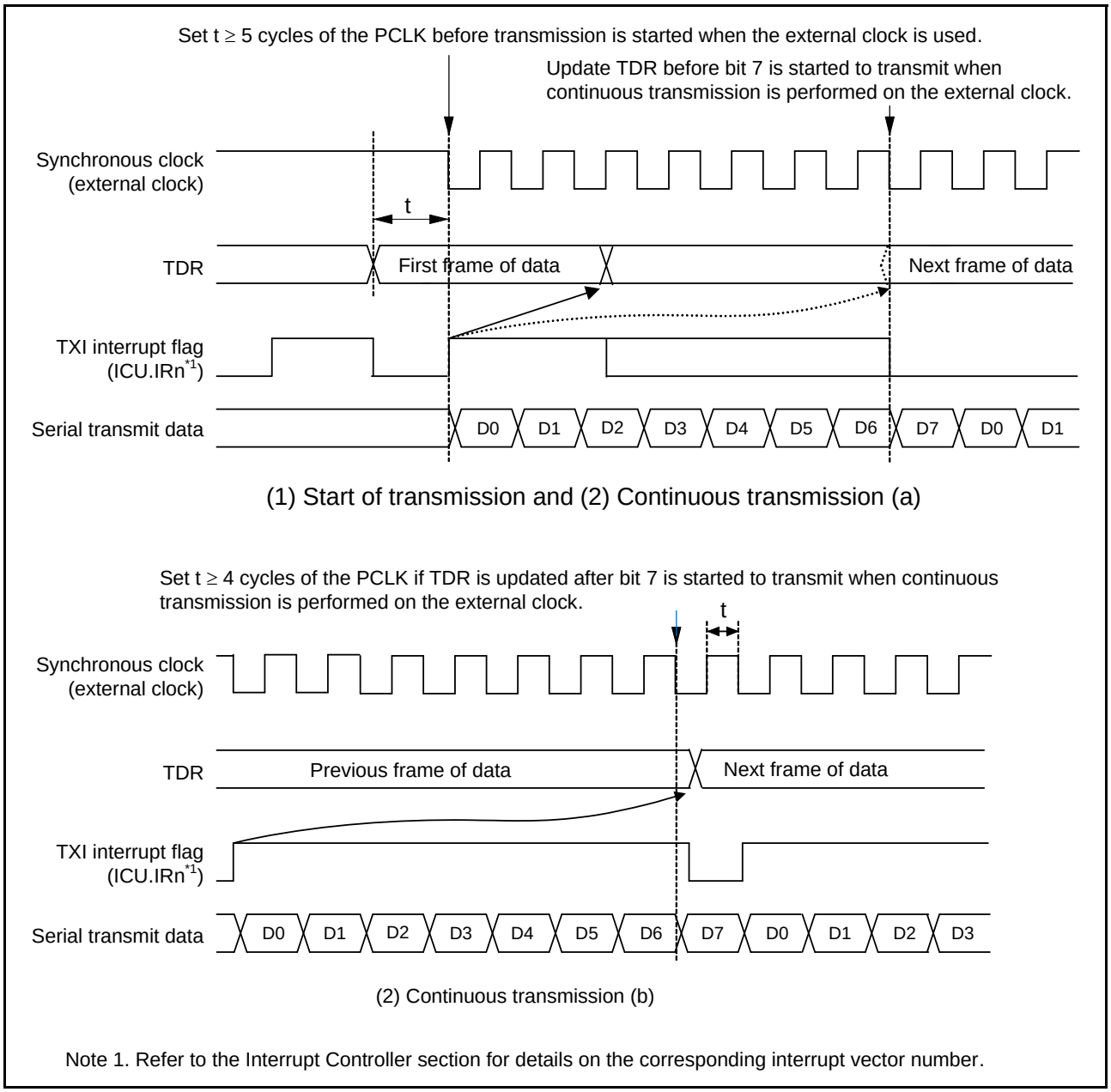



**Figure 26.73 Restrictions on Use of External Clock in Clock Synchronous Transmission**

### 26.13.7 Restrictions on Using DTC

When using the DTC to read RDR, be sure to set the receive data full interrupt (RXI) as the activation source of the relevant SCI.

### 26.13.8 Points to Note on Starting Transfer

At the point where transfer starts when the interrupt status flag (IRn.IR bit) in the interrupt controller is 1, follow the procedure below to clear interrupt requests before permitting operations (by setting the SCR.TE or SCR.RE bit to 1). For details on the interrupt status flag, refer to section 14, Interrupt Controller (ICUb).

- Confirm that transfer has stopped (the setting of the SCR.TE or SCR.RE bits is 0).
- Set the corresponding interrupt enable bit (SCR.TIE or SCR.RIE) to 0.
- Read the corresponding interrupt enable bit (SCR.TIE or SCR.RIE bit) to check that it has actually become 0.
- Set the interrupt status flag (IRn.IR bit) in the interrupt controller to 0.

### 26.13.9 SCI Operations during Low Power Consumption State

#### (1) Transmission

When making settings for the module stopped state or in transitions to software standby, stop operations (by setting the TIE, TE, and TEIE bits in the SCR to 0) after switching the TXDn pin to the general I/O port pin function. Clearing the TE bit to 0 resets the TSR and the TEND bit in the SSR. Depending on the port settings, output pins may output the level before a transition to the low power consumption state is made after release from the module stopped state or software standby mode. When transitions to these states are made during transmission, the data being transmitted become indeterminate.

To transmit data in the same transmission mode after cancellation of the low power consumption state, set the TE bit to 1, read SSR, and write data to TDR sequentially to start data transmission. To transmit data with a different transmission mode, initialize the SCI first.

Figure 26.74 shows a sample flowchart for transition to software standby mode during transmission. Figure 26.75 and Figure 26.76 show the port pin states during transition to software standby mode.

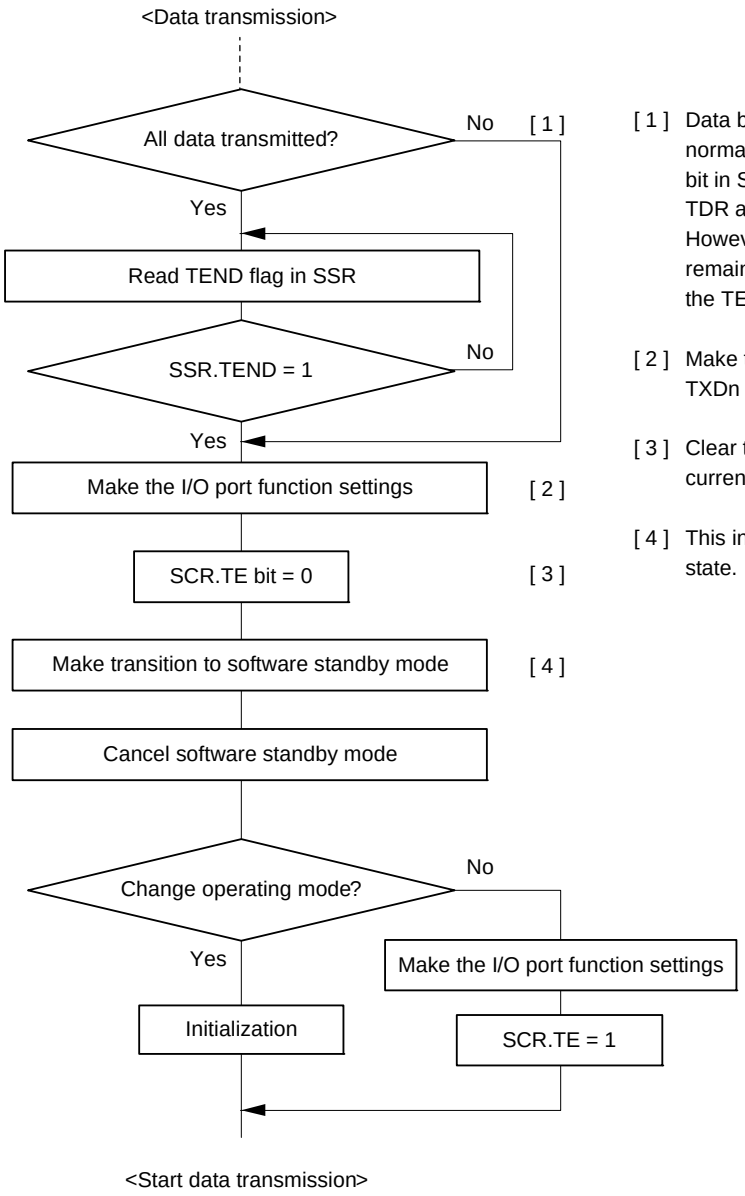
Before specifying the module stop state or making a transition to software standby mode from the transmission mode using DTC transfer, stop the transmit operations (TE = 0). To start transmission after cancellation using the DTC, set the TE bit to 1. The TXI interrupt flag is set to 1 and transmission starts using the DTC.

#### (2) Reception

Before specifying the module stop state or making a transition to software standby mode, stop the receive operations (RE = 0 in SCR). If transition is made during data reception, the data being received will be invalid.

To receive data in the same reception mode after cancellation of the low power consumption state, set the RE bit to 1, and then start reception. To receive data in a different reception mode, initialize the SCI first.

Figure 26.77 shows a sample flowchart for transition to software standby mode during reception.




[ 1 ] Data being transmitted is lost halfway. Data can be normally transmitted from the CPU by setting the TE bit in SCR to 1, reading SSR, and writing data to TDR after canceling software standby mode. However, if the DTC has been activated, the data remaining in the DTC will be transmitted when both the TE and TIE bits in SCR are set to 1.

[ 2 ] Make the I/O port function settings to switch the TXDn pin to operate as a general I/O port.

[ 3 ] Clear the TIE and TEIE bits in the SCR if they are currently set to 1.

[ 4 ] This includes the setting for the module stopped state.

**Figure 26.74 Example of Flowchart for Transition to Software Standby Mode during Transmission**

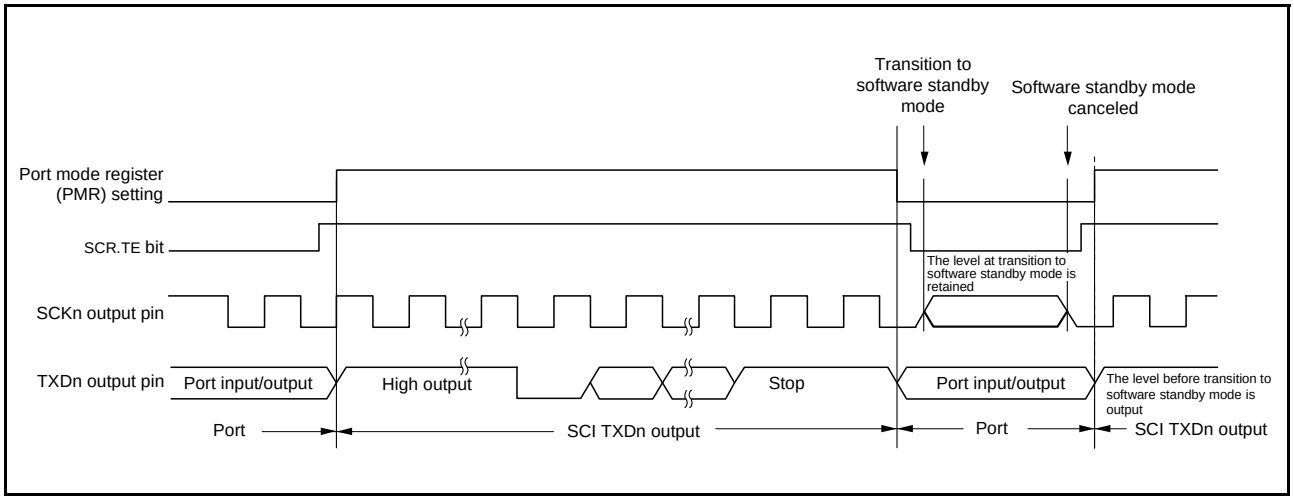



**Figure 26.75 Port Pin States during Transition to Software Standby Mode (Internal Clock, Asynchronous Transmission)**

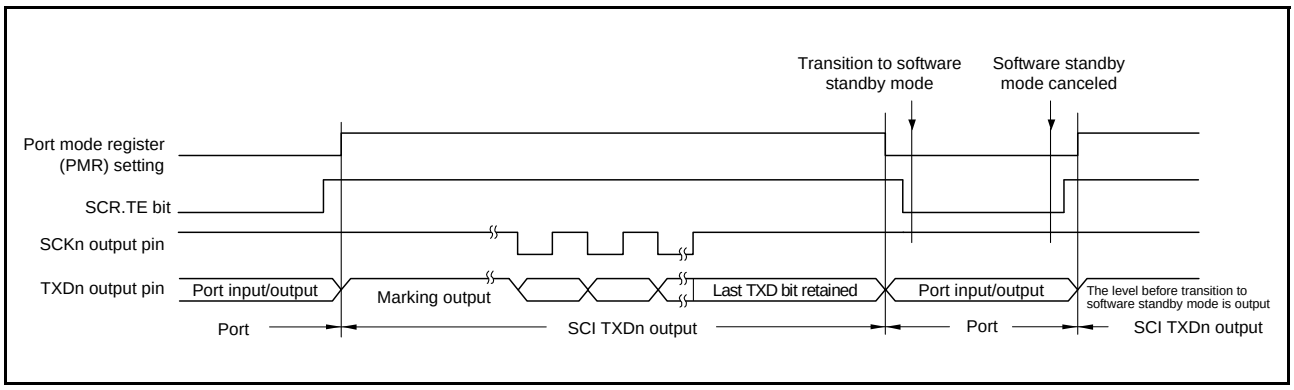



**Figure 26.76 Port Pin States during Transition to Software Standby Mode (Internal Clock, Clock Synchronous Transmission)**

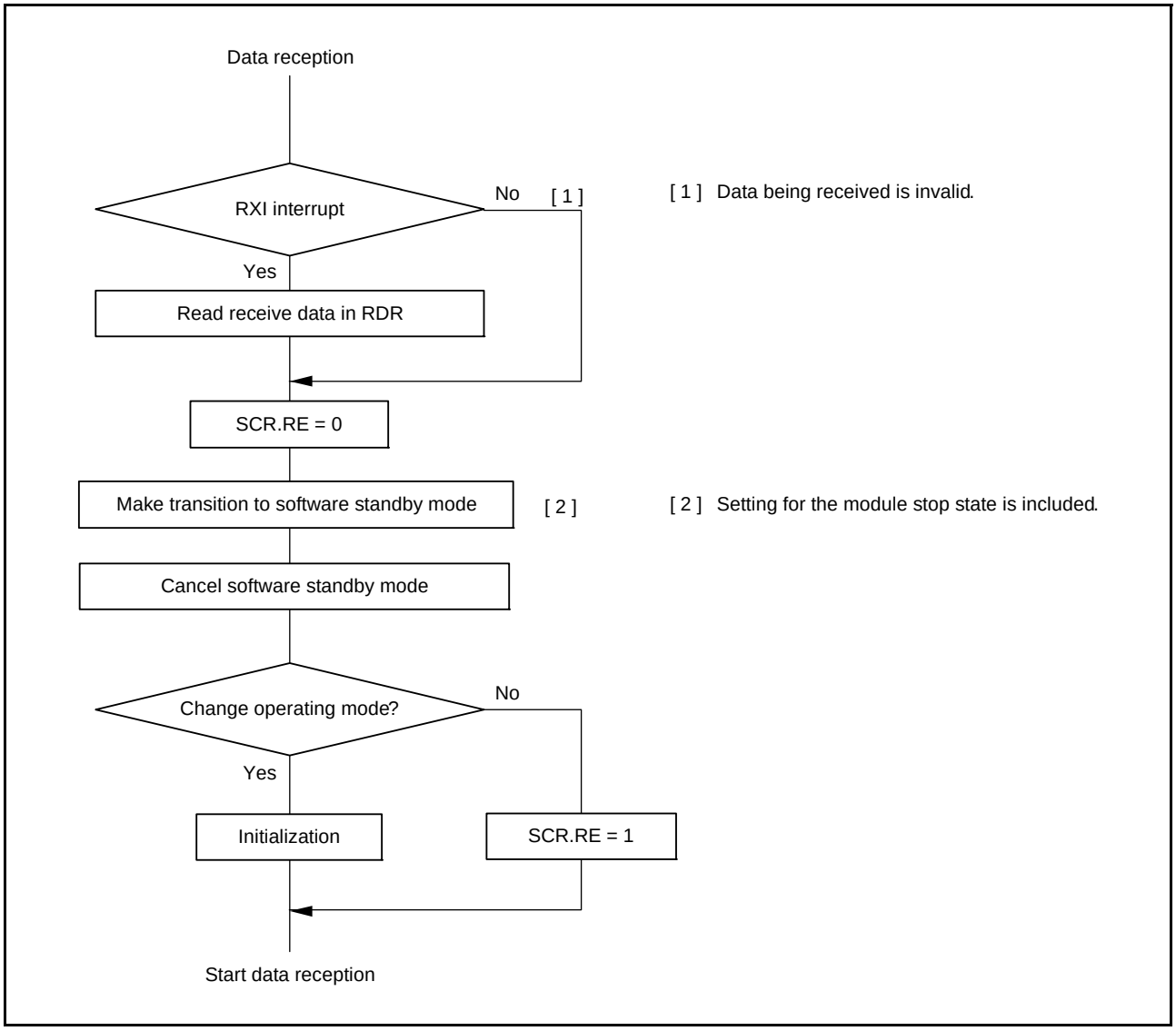



**Figure 26.77 Example of Flowchart for Transition to Software Standby Mode during Reception**

## 26.13.10 External Clock Input in Clock Synchronous Mode and Simple SPI Mode

In clock synchronous mode and simple SPI mode, the external clock SCKn must be input as follows:
High-pulse period, low-pulse period = 2 PCLK cycles or more, period = 6 PCLK cycles or more

## 26.13.11 Limitations on Simple SPI Mode

### (1) Master Mode

- Use a resistor to pull up or pull down the clock line matching the initial settings for the transfer clock set by the SPMR.CKPH and CKPOL bits when the SPMR.SSE bit is 1.
  This prevents the clock line from being placed in the high-impedance state when the SCR.TE bit is cleared to 0 or unexpected edges from being generated on the clock line when the SCR.TE bit is changed from 0 to 1. When the SPMR.SSE bit is 0 in single master mode, pulling up or pulling down the clock line is not necessary because the clock line is not placed in the high-impedance state even when the SCR.TE bit is cleared to 0.
- In the case of the setting for clock delay (the SPMR.CKPH bit is 1), the receive data full interrupt (RXI) is generated before the final clock edge on the SCKn pin as indicated in Figure 26.78. If the TE and RE bits in the SCR become 0 at this time before the final edge of the clock signal on the SCKn pin, the SCKn pin is placed in the high-impedance state, so the width of the last clock pulse of the transfer clock is shortened. Furthermore, an RXI interrupt may lead to the input signal on the SSn# pin of a connected slave going to the high level before the final edge of the clock signal on the SCKn pin, leading to incorrect operation of the slave.
- When operation is in multi-master mode, take care because the SCKn pin output becomes high-impedance while the input on the SSn# pin is at the low level if a mode-fault error occurs as the current character is being transferred, stopping supply of the clock signal to the connected slave. Remake the settings for the connected slave to avoid misaligned bits when transfer is restarted.

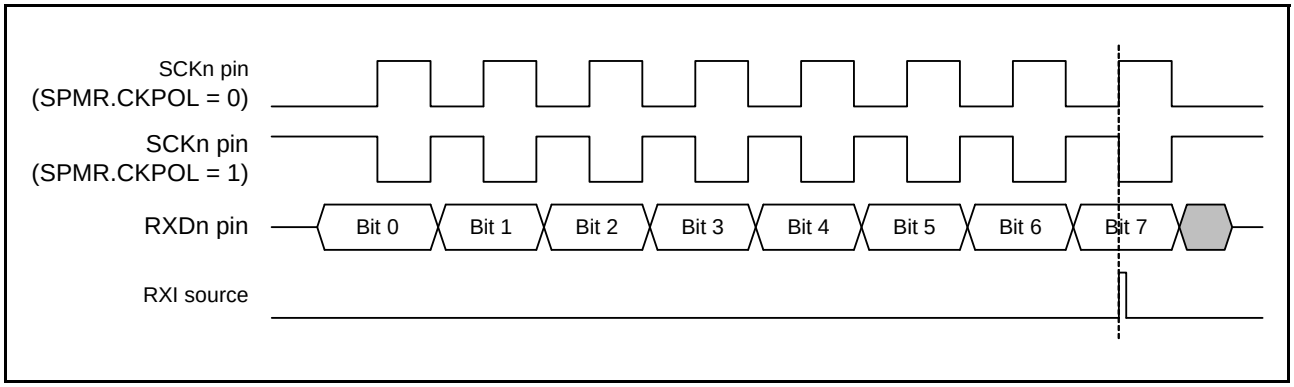



**Figure 26.78 Timing of the RXI Interrupt in Simple SPI Mode (with Clock Delay)**

### (2) Slave Mode

- Provide an external clock signal to the master the same as the data length for transfer.
- Control the input on the SSn# pin before the start and after the end of data transfer.
- When the level being input on the SSn# pin is to be changed from low to high while the current character is being transferred, set the TE and RE bits in the SCR to 0 and, after remaking the settings, restart transfer of the first byte.

## 26.13.12 Limitation 1 on Usage of the Extended Serial Mode Control Section

When the SHARPS bit in PCR is set to 1, output on the TXDX12/RXDX12 pin is only possible when the following conditions apply.

- The timer of the SCIf module is in Break Field low width output mode and the value of the TCST bit in TCR is 1 (when the TCST bit is set to 1, the high level continues to be output for up to one cycle of the clock source for counting by the timer counter before output of the low level)
- The value of the TE bit in SCI12.SCR is 1.

## 26.13.13 Limitation 2 on Usage of the Extended Serial Mode Control Section

An SCIe interrupt request is generated even if the extended serial mode is enabled. However, the SCIe interrupt should not be used during reception of a Start Frame because SCIf uses an SCIe interrupt request.
The two ways of dealing with this are described below. When a reception error is detected, clear the error flag of the SCIe and initialize the control section of the SCIf.

(1) Set the SCR.RIE bit of the SCIe to 0 to disable the output of interrupt requests. Check the error flags in the SSR register for SCIe on completion of the reception of a Start Frame, because an ERI interrupt is not generated if a reception error occurs. After reception of the Start Frame is completed, set the SCR.RIE bit of the SCIe to 1 by the time the first byte of the Information Frame is received.

(2) Set the SCR.RIE bit of the SCIe to 1 to disable RXI interrupts and enable ERI interrupts for ICU.
Clear the IRn.IR flag to enable the acceptance of RXI interrupts by ICU by the time the first byte of the Information Frame is received after the completion of Start Frame reception.

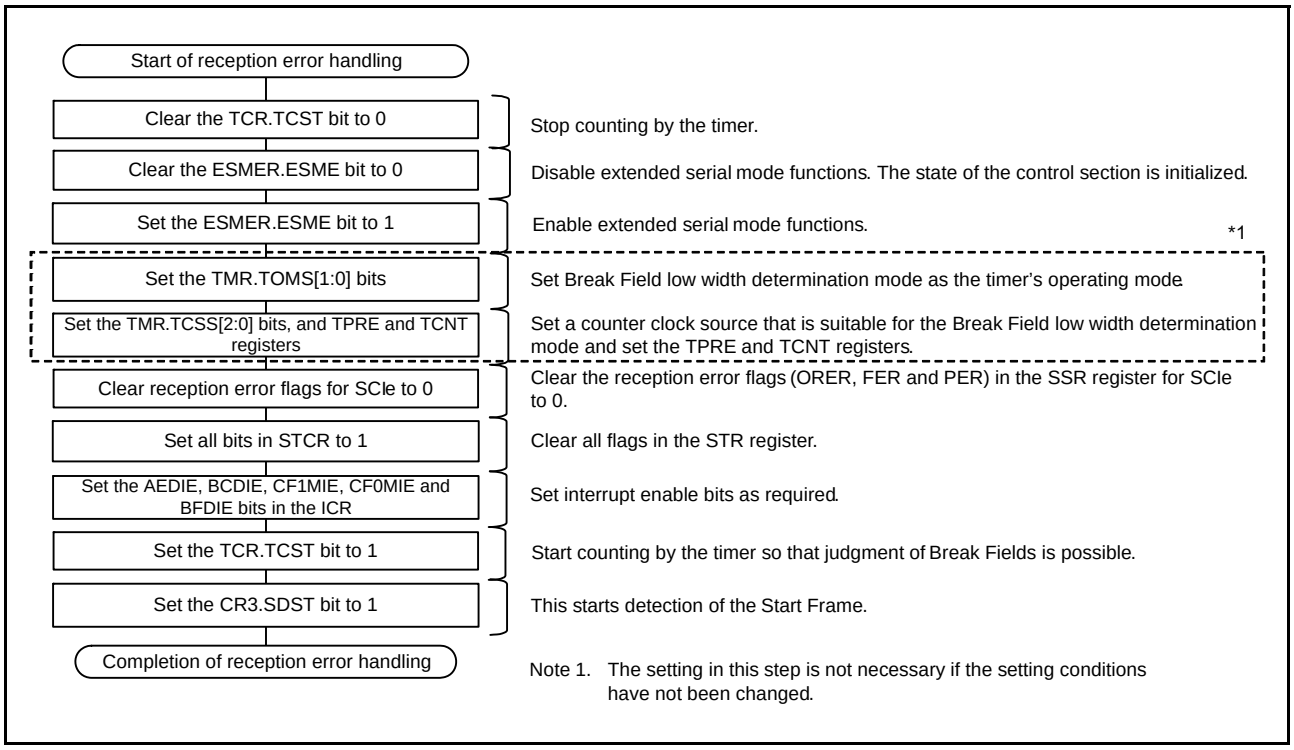



**Figure 26.79 Example of Flowchart for Reception Error Handling (during Reception of the Start Frame)**

# 27. I2C Bus Interface (RIIC)

This MCU has one I2C bus interface (RIIC module).
The RIIC module conforms with and provides a subset of the NXP I2C bus (Inter-IC-Bus) interface functions.

## 27.1 Overview

Table 27.1 lists the specifications of the RIIC, Figure 27.1 shows a block diagram of the RIIC, and Figure 27.2 shows an example of I/O pin connections to external circuits (I2C bus configuration example). Table 27.2 lists the I/O pins of the RIIC.

**Table 27.1 RIIC Specifications (1/2)**

| Item | Description |
|---|---|
| Communications format | • I2C bus format or SMBus format<br>• Master mode or slave mode selectable<br>• Automatic securing of the various set-up times, hold times, and bus-free times for the transfer rate |
| Transfer rate | Up to 400 kbps |
| SCL clock | For master operation, the duty cycle of the SCL clock is selectable in the range from 4% to 96%. |
| Issuing and detecting conditions | Start, restart, and stop conditions are automatically generated. Start conditions (including restart conditions) and stop conditions are detectable. |
| Slave address | • Up to three different slave addresses can be set.<br>• 7-bit and 10-bit address formats are supported (along with the use of both at once).<br>• General call addresses, device ID addresses, and SMBus host addresses are detectable. |
| Acknowledgement | • For transmission, the acknowledge bit is automatically loaded.<br>Transfer of the next data for transmission can be automatically suspended on detection of a not-acknowledge bit.<br>• For reception, the acknowledge bit is automatically transmitted.<br>If a wait between the eighth and ninth clock cycles has been selected, software control of the value in the acknowledge field in response to the received value is possible. |
| Wait function | • In reception, the following periods of waiting can be obtained by holding the SCL clock at the low level:<br>Waiting between the eighth and ninth clock cycles<br>Waiting between the ninth clock cycle and the first clock cycle of the next transfer |
| SDA output delay function | Timing of the output of transmitted data, including the acknowledge bit, can be delayed. |
| Arbitration | • For multi-master operation<br>Operation to synchronize the SCL clock in cases of conflict with the SCL signal from another master is possible.<br>When issuing the start condition would create conflict on the bus, loss of arbitration is detected by testing for non-matching between the internal signal for the SDA line and the level on the SDA line.<br>In master operation, loss of arbitration is detected by testing for non-matching between the signal on the SDA line and the internal signal for the SDA line.<br>• Loss of arbitration due to detection of the start condition while the bus is busy is detectable (to prevent the issuing of double start conditions).<br>• Loss of arbitration in transfer of a not-acknowledge bit due to the internal signal for the SDA line and the level on the SDA line not matching is detectable.<br>• Loss of arbitration due to non-matching of internal and line levels for data is detectable in slave transmission. |
| Timeout function | The internal timeout function is capable of detecting long-interval stop of the SCL clock. |
| Noise cancellation | The interface incorporates digital noise filters for both the SCL and SDA signals, and the width for noise cancellation by the filters is adjustable by software. |
| Interrupt sources | • Four sources:<br>Error in transfer or occurrence of events<br>Detection of arbitration, NACK, timeout, a start condition including a restart condition, or a stop condition<br>• Receive-data-full (including matching with a slave address)<br>• Transmit-data-empty (including matching with a slave address)<br>• Transmission complete |
| Low power consumption function | Module stop state can be set. |

**Table 27.1 RIIC Specifications (2/2)**

| Item | Description |
|---|---|
| RIIC operating modes | • Four<br>Master transmit mode, master receive mode, slave transmit mode, and slave receive mode |
| Event link function (output) | • Four sources:<br>Error in transfer or occurrence of events<br>Detection of arbitration, NACK, timeout, a start condition including a restart condition, or a stop condition<br>• Receive-data-full (including matching with a slave address)<br>• Transmit-data-empty (including matching with a slave address)<br>• Transmission complete |

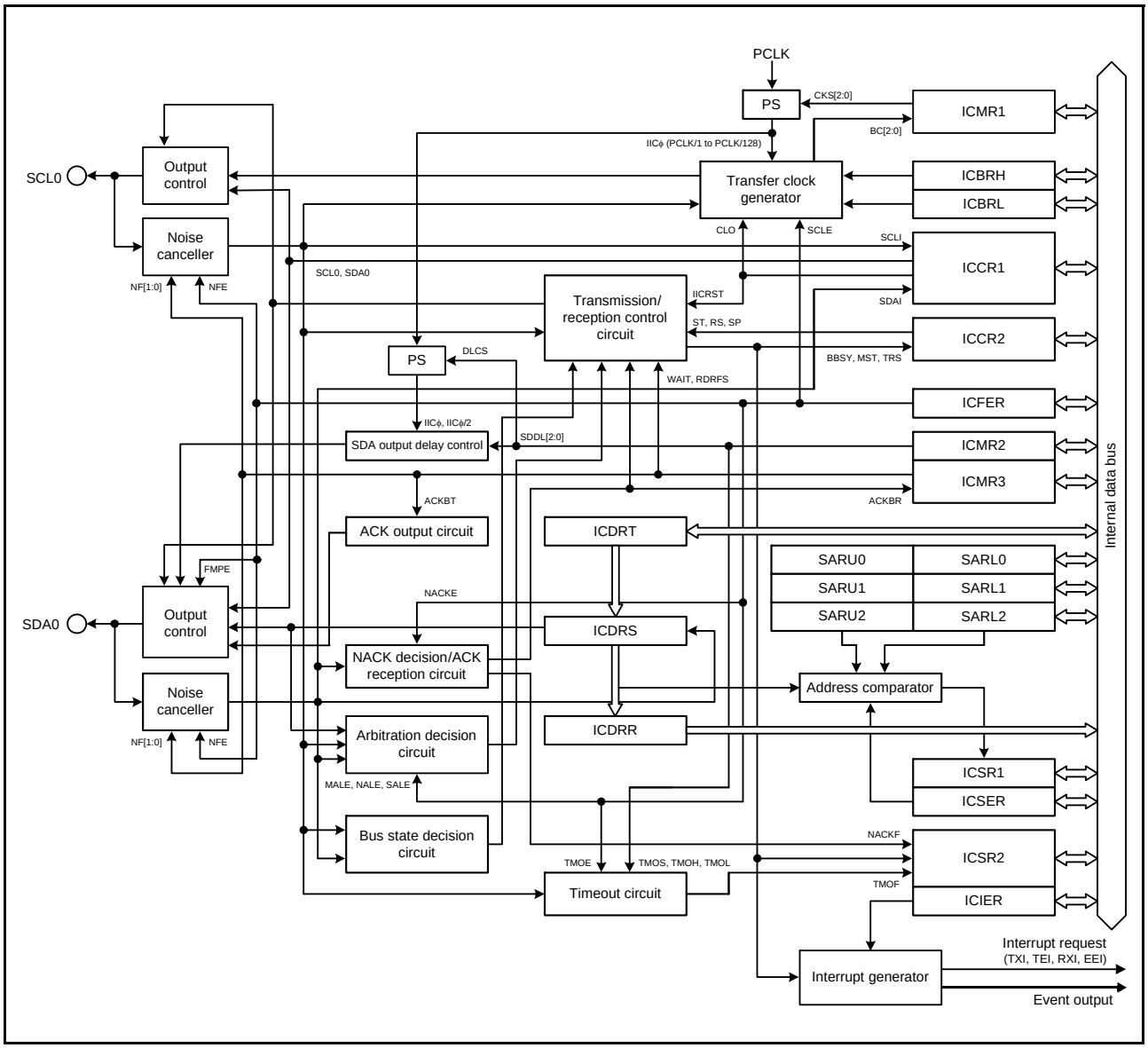



**Figure 27.1 RIIC Block Diagram**

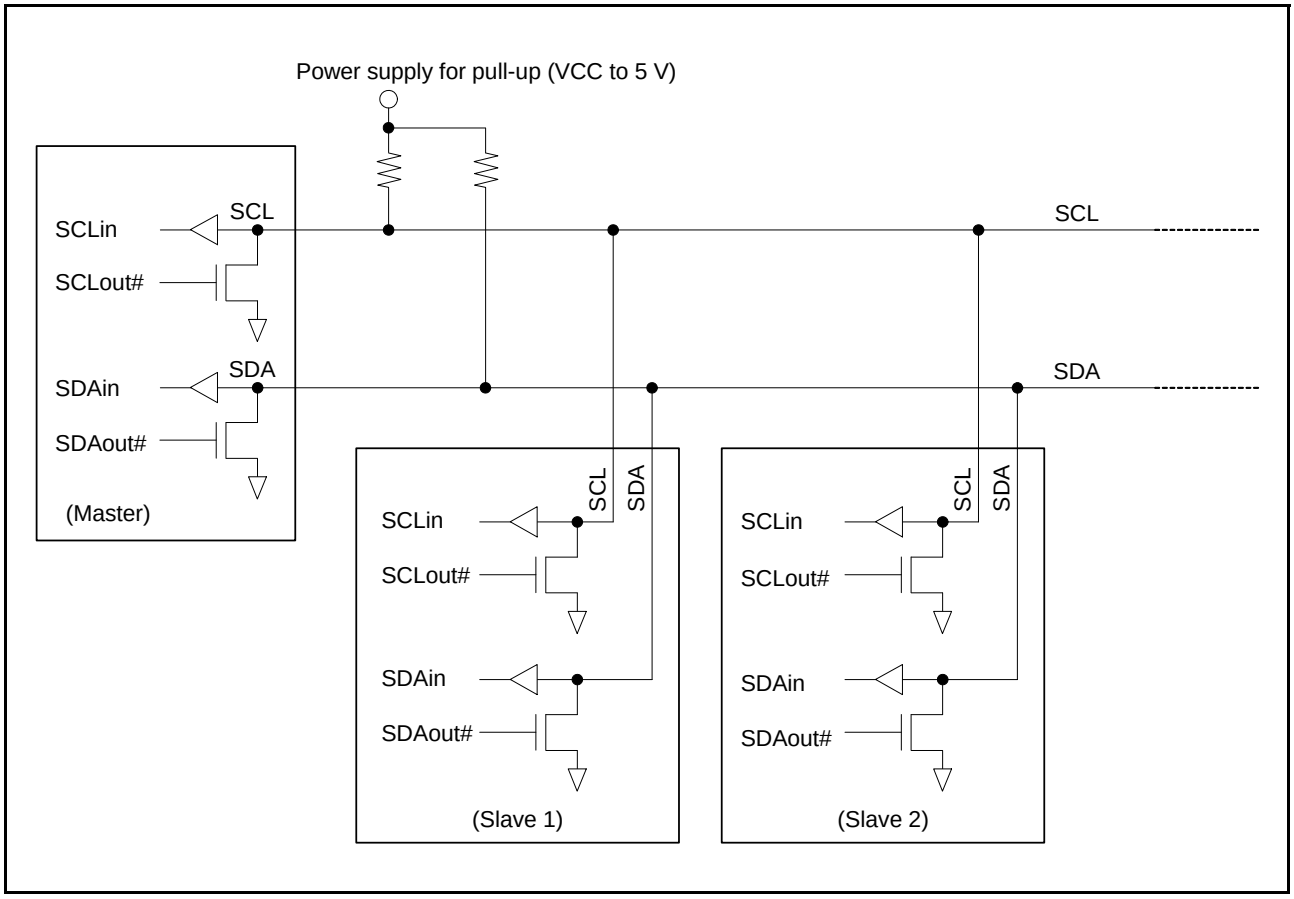



**Figure 27.2 Connections to the External Circuit by the I/O Pins (I2C Bus Configuration Example)**

The input level of the signals for RIIC is CMOS when I2C bus is selected (the ICMR3.SMBS bit = 0), or TTL when SMBus is selected (the ICMR3.SMBS bit = 1).

**Table 27.2 Pin Configuration**

| Channel | Pin Name | I/O | Function |
|---|---|---|---|
| RIIC0 | SCL0 | I/O | RIIC0 serial clock I/O pin |
| | SDA0 | I/O | RIIC0 serial data I/O pin |

## 27.2 Register Descriptions

### 27.2.1 I2C Bus Control Register 1 (ICCR1)

Address(es): RIIC0.ICCR1 0008 8300h

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | ICE | IICRST | CLO | SOWP | SCLO | SDAO | SCLI | SDAI |
| Value after reset: | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b0 | SDAI | SDA Line Monitor | 0: SDA0 line is low.<br>1: SDA0 line is high. | R |
| b1 | SCLI | SCL Line Monitor | 0: SCL0 line is low.<br>1: SCL0 line is high. | R |
| b2 | SDAO | SDA Output Control/Monitor | • Read:<br>0: The RIIC has driven the SDA0 pin low.<br>1: The RIIC has released the SDA0 pin.<br>• Write:<br>0: The RIIC drives the SDA0 pin low.<br>1: The RIIC releases the SDA0 pin. | R/W |
| b3 | SCLO | SCL Output Control/Monitor | • Read:<br>0: The RIIC has driven the SCL0 pin low.<br>1: The RIIC has released the SCL0 pin.<br>• Write:<br>0: The RIIC drives the SCL0 pin low.<br>1: The RIIC releases the SCL0 pin.<br>(High level output is achieved through an external pull-up resistor.) | R/W |
| b4 | SOWP | SCLO/SDAO Write Protect | 0: Bits SCLO and SDAO can be written.<br>1: Bits SCLO and SDAO are protected.<br>(This bit is read as 1.) | R/W |
| b5 | CLO | Extra SCL Clock Cycle Output | 0: Does not output an extra SCL clock cycle (default).<br>1: Outputs an extra SCL clock cycle.<br>(The CLO bit is cleared automatically after one clock cycle is output.) | R/W |
| b6 | IICRST | I2C Bus Interface Internal Reset | 0: Clears the RIIC reset or internal reset.<br>1: Initiates the RIIC reset or internal reset.<br>(Clears the bit counter and the SCL0/SDA0 output latch) | R/W |
| b7 | ICE | I2C Bus Interface Enable | 0: Disable (SCL0 and SDA0 pins in inactive state)<br>1: Enable (SCL0 and SDA0 pins in active state)<br>(Combined with the IICRST bit to select either RIIC or internal reset.) | R/W |

**SDAO Bit (SDA Output Control/Monitor) and SCLO Bit (SCL Output Control/Monitor)**

These bits are used to directly control the SDA0 and SCL0 signals output from the RIIC.

When writing to these bits, also write 0 to the SOWP bit.

The result of setting these bits is input to the RIIC via the input buffer. When slave mode is selected, a START condition may be detected and the bus may be released depending on the bit settings.

Do not rewrite these bits during a START condition, STOP condition, repeated START condition, or during transmission or reception. Operation after rewriting under the above conditions is not guaranteed.

When reading these bits, the state of signals output from the RIIC can be read.

### CLO Bit (Extra SCL Clock Cycle Output)

This bit is used to output an extra SCL clock cycle for debugging or error processing.
Normally, set the bit to 0. Setting the bit to 1 in a normal communication state causes a communication error.
For details on this function, refer to section 27.11.2, Extra SCL Clock Cycle Output Function.

### IICRST Bit (I2C Bus Interface Internal Reset)

This bit is used to reset the internal states of the RIIC.
Setting this bit to 1 initiates an RIIC reset or internal reset.
Whether an RIIC reset or internal reset is initiated is determined according to the combination with the ICE bit. Table 27.3 lists the resets of the RIIC.
The RIIC reset resets all registers and internal states of the RIIC, and the internal reset resets the bit counter (BC[2:0] bits in ICMR1), the I2C bus shift register (ICDRS), and the I2C bus status registers (ICSR1 and ICSR2) as well as the internal states of the RIIC. For the reset conditions for each register, refer to section 27.14, Resets and Register and Function States When Issuing Each Condition.
An internal reset initiated with the IICRST bit set to 1 during operation (with the ICE bit set to 1) resets the internal states of the RIIC without initializing the port settings and the control and setting registers of the RIIC when the bus or RIIC hangs up due to a communication error.
If the RIIC hangs up in a low level output state, resetting the internal states cancels the low level output state and releases the bus with the SCL0 pin and SDA0 pin at a high impedance.

Note: • If an internal reset is initiated using the IICRST bit for a bus hang-up occurred during communication with the master device in slave mode, the states may become different between the slave device and the master device (due to the difference in the bit counter information). For this reason, do not initiate an internal reset in slave mode, but initiate restoration processing from the master device. If an internal reset is necessary because the RIIC hangs up with the SCL0 line in a low level output state in slave mode, initiate an internal reset and then issue a restart condition from the master device or resume communication from the start condition issuance after issuing a stop condition. If communication is restarted by initiating a reset solely in the slave device without issuing a start condition or restart condition from the master device, synchronization will be lost because the master and slave devices operate asynchronously.

**Table 27.3 RIIC Resets**

| IICRST | ICE | State | Specifications |
|---|---|---|---|
| 1 | 0 | RIIC reset | Resets all registers and internal states of the RIIC. |
| | 1 | Internal reset | Reset the ICMR1.BC[2:0] bits, the ICSR1, ICSR2, and ICDRS registers, and the internal states of the RIIC. |

### ICE Bit (I2C Bus Interface Enable)

This bit selects the active or inactive state of the SCL0 and SDA0 pins. It can also be combined with the IICRST bit to initiate two types of resets. See Table 27.3, RIIC Resets, for the types of resets.
Set the ICE bit to 1 when using the RIIC. The SCL0 and SDA0 pins are placed in the active state when the ICE bit is set to 1.
Set the ICE bit to 0 when the RIIC is not to be used. The SCL0 and SDA0 pins are placed in the inactive state when the ICE bit is set to 0. Do not assign the SCL0 or SDA0 pin to the RIIC when setting up the multi-function pin controller (MPC). Note that the slave address comparison operation is carried out if the pins are assigned to the RIIC.

## 27.2.2 I2C Bus Control Register 2 (ICCR2)

Address(es): RIIC0.ICCR2 0008 8301h

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | BBSY | MST | TRS | — | SP | RS | ST | — |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b0 | — | Reserved | This bit is read as 0. The write value should be 0. | R/W |
| b1 | ST | Start Condition Issuance Request | 0: Does not request to issue a start condition.<br>1: Requests to issue a start condition. | R/W |
| b2 | RS | Restart Condition Issuance Request | 0: Does not request to issue a restart condition.<br>1: Requests to issue a restart condition. | R/W |
| b3 | SP | Stop Condition Issuance Request | 0: Does not request to issue a stop condition.<br>1: Requests to issue a stop condition. | R/W |
| b4 | — | Reserved | This bit is read as 0. The write value should be 0. | R/W |
| b5 | TRS | Transmit/Receive Mode | 0: Receive mode<br>1: Transmit mode | R/W*1 |
| b6 | MST | Master/Slave Mode | 0: Slave mode<br>1: Master mode | R/W*1 |
| b7 | BBSY | Bus Busy Detection Flag | 0: The I2C bus is released (bus free state).<br>1: The I2C bus is occupied (bus busy state). | R |

Note 1. When the MTWP bit in ICMR1 is set to 1, the MST and TRS bits can be written to.

**ST Bit (Start Condition Issuance Request)**

This bit is used to request transition to master mode and issuance of a start condition.

When this bit is set to 1 to request to issue a start condition, a start condition is issued when the BBSY flag is set to 0 (bus free state).

For details on the start condition issuance, refer to section 27.10, Start Condition/Restart Condition/Stop Condition Issuing Function.

[Setting condition]

- When 1 is written to the ST bit

[Clearing conditions]

- When 0 is written to the ST bit
- When a start condition has been issued (A start condition is detected)
- When the AL (arbitration-lost) flag in ICSR2 is set to 1
- When 1 is written to the IICRST bit in ICCR1 to apply an RIIC reset or an internal reset

Note: • Set the ST bit to 1 (start condition issuance request) when the BBSY flag is set to 0 (bus free state).
Note that arbitration may be lost if the ST bit is set to 1 (start condition issuance request) when the BBSY flag is set to 1 (bus busy state).

### RS Bit (Restart Condition Issuance Request)

This bit is used to request that a restart condition be issued in master mode.
When this bit is set to 1 to request to issue a restart condition, a restart condition is issued when the BBSY flag is set to 1 (bus busy state) and the MST bit is set to 1 (master mode).
For details on the restart condition issuance, refer to section 27.10, Start Condition/Restart Condition/Stop Condition Issuing Function.
[Setting condition]

- When 1 is written to the RS bit with the BBSY flag in ICCR2 set to 1

[Clearing conditions]

- When 0 is written to the RS bit
- When a restart condition has been issued (A start condition is detected)
- When the AL (arbitration-lost) flag in ICSR2 is set to 1
- When 1 is written to the IICRST bit in ICCR1 to apply an RIIC reset or an internal reset

Note: • Do not set the RS bit to 1 while issuing a stop condition.
Note: • If 1 (requests to issue a restart condition) is written to the RS bit in slave mode, the restart condition is not issued but the RS bit remains set to 1.
If the operating mode changes to master mode with the bit not being cleared, note that the restart condition may be issued.

### SP Bit (Stop Condition Issuance Request)

This bit is used to request that a stop condition be issued in master mode.
When this bit is set to 1 to request to issue a stop condition, a stop condition is issued when the BBSY flag is set to 1 (bus busy state) and the MST bit is set to 1 (master mode).
For details on the stop condition issuance, refer to section 27.10, Start Condition/Restart Condition/Stop Condition Issuing Function.
[Setting condition]

- When 1 is written to the SP bit with both the BBSY flag and the MST bit in ICCR2 set to 1

[Clearing conditions]

- When 0 is written to the SP bit
- When a stop condition has been issued (A stop condition is detected)
- When the AL (arbitration-lost) flag in ICSR2 is set to 1
- When a start condition and a restart condition are detected
- When 1 is written to the IICRST bit in ICCR1 to apply an RIIC reset or an internal reset

Note: • Writing to the SP bit is not possible while the setting of the BBSY flag is 0 (bus free state).
Note: • Do not set the SP bit to 1 while a restart condition is being issued.

### TRS Bit (Transmit/Receive Mode)

This bit indicates transmit or receive mode.
The RIIC is in receive mode when the TRS bit is set to 0 and is in transmit mode when the bit is set to 1. Combination of this bit and the MST bit indicates the operating mode of the RIIC.
The value of TRS bit is automatically changed to 1 for transmit mode or 0 for receive mode by issuing or detection of a start condition and setting or clearing of the R/W# bit. Although writing to the TRS bit is possible when the MTWP bit in ICMR1 is set to 1, writing to this bit is not necessary during normal usage.
[Setting conditions]

- When a start condition is issued normally according to the start condition issuance request (when a start condition is detected with the ST bit set to 1)
- When the R/W# bit added to the slave address is set to 0 in master mode
- When the address received in slave mode matches the address enabled in ICSER, with the R/W# bit set to 1
- When 1 is written to the TRS bit with the MTWP bit in ICMR1 set to 1

[Clearing conditions]

- When a stop condition is detected
- The AL (arbitration-lost) flag in ICSR2 being set to 1
- In master mode, reception of a slave address to which an R/W# bit with the value 1 is appended
- In slave mode, a match between the received address and the address enabled in ICSER when the value of the received R/W# bit is 0 (including cases where the received address is the general call address)
- In slave mode, a restart condition is detected (a start condition is detected with ICCR2.BBSY = 1 and ICCR2.MST = 0)
- When 0 is written to the TRS bit with the MTWP bit in ICMR1 set to 1
- When 1 is written to the IICRST bit in ICCR1 to apply an RIIC reset or an internal reset

### MST Bit (Master/Slave Mode)

This bit indicates master or slave mode.
The RIIC is in slave mode when the MST bit is set to 0 and is in master mode when the bit is set to 1. Combination of this bit and the TRS bit indicates the operating mode of the RIIC.
The value of the MST bit is automatically changed to 1 for master mode or 0 for slave mode by issuing of a start condition and issuing or detection of a stop condition, etc. Although writing to the MST bit is possible when the MTWP bit in ICMR1 is set to 1, writing to this bit is not necessary during normal usage.
[Setting conditions]

- When a start condition is issued normally according to the start condition issuance request (when a start condition is detected with the ST bit set to 1)
- When 1 is written to the MST bit with the MTWP bit in ICMR1 set to 1

[Clearing conditions]

- When a stop condition is detected
- When the AL (arbitration-lost) flag in ICSR2 is set to 1
- When 0 is written to the MST bit with the MTWP bit in ICMR1 set to 1
- When 1 is written to the IICRST bit in ICCR1 to apply an RIIC reset or an internal reset

**BBSY Flag (Bus Busy Detection Flag)**

The BBSY flag indicates whether the I2C bus is occupied (bus busy state) or released (bus free state).
This bit is set to 1 when the SDA0 line changes from high to low under the condition of SCL0 line = high, assuming that a start condition has been issued.
When the SDA0 line changes from low to high under the condition of SCL0 line = high, this bit is cleared to 0 after the bus free time (specified in ICBRL) start condition is not detected, assuming that a stop condition has been issued.
[Setting condition]

- When a start condition is detected

[Clearing conditions]

- When the bus free time (specified in ICBRL) start condition is not detected after detecting a stop condition
- When 1 is written to the IICRST bit in ICCR1 with the ICE bit in ICCR1 set to 0 (RIIC reset)

## 27.2.3 I2C Bus Mode Register 1 (ICMR1)

Address(es): RIIC0.ICMR1 0008 8302h

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | MTWP | CKS[2:0] | | | BCWP | BC[2:0] | | |
| Value after reset: | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b2 to b0 | BC[2:0] | Bit Counter | b2 b0<br>0 0 0: 9 bits<br>0 0 1: 2 bits<br>0 1 0: 3 bits<br>0 1 1: 4 bits<br>1 0 0: 5 bits<br>1 0 1: 6 bits<br>1 1 0: 7 bits<br>1 1 1: 8 bits | R/W*1 |
| b3 | BCWP | BC Write Protect | 0: Enables a value to be written in the BC[2:0] bits.<br>(This bit is read as 1.) | R/W*1 |
| b6 to b4 | CKS[2:0] | Internal Reference Clock Selection | Selects the internal reference clock source (IICϕ) for the RIIC.<br>b6 b4<br>0 0 0: PCLK/1 clock<br>0 0 1: PCLK/2 clock<br>0 1 0: PCLK/4 clock<br>0 1 1: PCLK/8 clock<br>1 0 0: PCLK/16 clock<br>1 0 1: PCLK/32 clock<br>1 1 0: PCLK/64 clock<br>1 1 1: PCLK/128 clock | R/W |
| b7 | MTWP | MST/TRS Write Protect | 0: Disables writing to the MST and TRS bits in ICCR2.<br>1: Enables writing to the MST and TRS bits in ICCR2. | R/W |

Note 1. Rewrite the BC[2:0] bits and clear the BCWP bit to 0 at the same time.

### BC[2:0] Bits (Bit Counter)

These bits function as a counter that indicates the number of bits remaining to be transferred at the detection of a rising edge on the SCL0 line. Although these bits are writable and readable, it is not necessary to access these bits under normal conditions.

To write to these bits, specify the number of bits to be transferred plus one (data is transferred with an additional acknowledge bit) between transferred frames when the SCL0 line is at a low level.

The values of the BC[2:0] bits return to 000b at the end of a data transfer including the acknowledge bit or when a start condition including a restart condition is detected.

## 27.2.4 I2C Bus Mode Register 2 (ICMR2)

Address(es): RIIC0.ICMR2 0008 8303h

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | DLCS | SDDL[2:0] | | | TMWE | TMOH | TMOL | TMOS |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b0 | TMOS | Timeout Detection Time Selection | 0: Long mode is selected.<br>1: Short mode is selected. | R/W |
| b1 | TMOL | Timeout L Count Control | 0: Count is disabled while the SCL0 line is at a low level.<br>1: Count is enabled while the SCL0 line is at a low level. | R/W |
| b2 | TMOH | Timeout H Count Control | 0: Count is disabled while the SCL0 line is at a high level.<br>1: Count is enabled while the SCL0 line is at a high level. | R/W |
| b3 | TMWE | Timeout Internal Counter Write Enable | 0: Writing to internal counter of timeout detection function is disabled.<br>1: Writing to internal counter of timeout detection function is enabled. | R/W |
| b6 to b4 | SDDL[2:0] | SDA Output Delay Counter | • When ICMR2.DLCS = 0 (IICϕ)<br>b6 b4<br>0 0 0: No output delay<br>0 0 1: 1 IICϕ cycle<br>0 1 0: 2 IICϕ cycles<br>0 1 1: 3 IICϕ cycles<br>1 0 0: 4 IICϕ cycles<br>1 0 1: 5 IICϕ cycles<br>1 1 0: 6 IICϕ cycles<br>1 1 1: 7 IICϕ cycles<br>• When ICMR2.DLCS = 1 (IICϕ/2)<br>b6 b4<br>0 0 0: No output delay<br>0 0 1: 1 or 2 IICϕ cycles<br>0 1 0: 3 or 4 IICϕ cycles<br>0 1 1: 5 or 6 IICϕ cycles<br>1 0 0: 7 or 8 IICϕ cycles<br>1 0 1: 9 or 10 IICϕ cycles<br>1 1 0: 11 or 12 IICϕ cycles<br>1 1 1: 13 or 14 IICϕ cycles | R/W |
| b7 | DLCS | SDA Output Delay Clock Source Selection | 0: The internal reference clock (IICϕ) is selected as the clock source of the SDA output delay counter.<br>1: The internal reference clock divided by 2 (IICϕ/2) is selected as the clock source of the SDA output delay counter.*1 | R/W |

Note 1. The setting DLCS = 1 (IICϕ/2) only becomes valid when SCL is at the low level. When SCL is at the high level, the setting DLCS = 1 becomes invalid and the clock source becomes the internal reference clock (IICϕ).

### TMOS Bit (Timeout Detection Time Selection)

This bit is used to select long mode or short mode for the timeout detection time when the timeout function is enabled (TMOE bit = 1 in ICFER). When this bit is set to 0, long mode is selected. When this bit is set to 1, short mode is selected. In long mode, the timeout detection internal counter functions as a 16 bit-counter. In short mode, the counter functions as a 14 bit-counter. While the SCL0 line is in the state that enables this counter as specified by bits TMOH and TMOL, the counter counts up in synchronization with the internal reference clock (IICϕ) as a count source.
For details on the timeout function, refer to section 27.11.1, Timeout Function.

### TMOL Bit (Timeout L Count Control)

This bit is used to enable or disable the internal counter of the timeout function to count up while the SCL0 line is held low when the timeout function is enabled (TMOE bit = 1 in ICFER).

### TMOH Bit (Timeout H Count Control)

This bit is used to enable or disable the internal counter of the timeout function to count up while the SCL0 line is held high when the timeout function is enabled (TMOE bit = 1 in ICFER).

### TMWE Bit (Timeout Internal Counter Write Enable)

This bit is used to select whether or not to allocate the timeout internal counter (TMOCNTL/TMOCNTU) to the address of the slave address register (SARL0/SARU0).
When this bit is set to 1, the addresses of the timeout internal counters (TMOCNTL and TMOCNTU) are allocated to the addresses of SARL0 and SARU0.

### SDDL[2:0] Bits (SDA Output Delay Counter)

The SDA output can be delayed by the SDDL[2:0] setting. This counter works with the clock source selected by the DLCS bit. The setting of this function can be used for all types of SDA output, including the transmission of the acknowledge bit.
Set the SDA output delay time to meet the I2C bus standard (within the data enable time/acknowledge enable time*1) or the SMBus standard (within the data hold time: 300 ns or more, and SCL-clock low-level period - the data setup time: 250 ns). Note that, if a value outside the standard is set, communication with communication devices may malfunction or it may seemingly become a start condition or stop condition depending on the bus state.
For details on this function, refer to section 27.5, Facility for Delaying SDA Output.

Note 1. Data enable time/acknowledge enable time
3,450 ns (up to 100 kbps: standard mode [Sm])
900 ns (up to 400 kbps: fast mode [Fm])

## 27.2.5 I2C Bus Mode Register 3 (ICMR3)

Address(es): RIIC0.ICMR3 0008 8304h

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | SMBS | WAIT | RDRFS | ACKWP | ACKBT | ACKBR | NF[1:0] | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b1, b0 | NF[1:0] | Noise Filter Stage Selection | b1 b0<br>0 0: Noise of up to one IICϕ cycle is filtered out (single-stage filter).<br>0 1: Noise of up to two IICϕ cycles is filtered out (2-stage filter).<br>1 0: Noise of up to three IICϕ cycles is filtered out (3-stage filter).<br>1 1: Noise of up to four IICϕ cycles is filtered out (4-stage filter). | R/W |
| b2 | ACKBR | Receive Acknowledge | 0: A 0 is received as the acknowledge bit (ACK reception).<br>1: A 1 is received as the acknowledge bit (NACK reception). | R |
| b3 | ACKBT | Transmit Acknowledge | 0: A 0 is sent as the acknowledge bit (ACK transmission).<br>1: A 1 is sent as the acknowledge bit (NACK transmission). | R/W*1 |
| b4 | ACKWP | ACKBT Write Protect | 0: Modification of the ACKBT bit is disabled.<br>1: Modification of the ACKBT bit is enabled. | W*1 |
| b5 | RDRFS | RDRF Flag Set Timing Selection | 0: The RDRF flag is set at the rising edge of the ninth SCL clock cycle. (The SCL0 line is not held low at the falling edge of the eighth clock cycle.)<br>1: The RDRF flag is set at the rising edge of the eighth SCL clock cycle. (The SCL0 line is held low at the falling edge of the eighth clock cycle.) Low-hold is released by writing a value to the ACKBT bit. | R/W*2 |
| b6 | WAIT | WAIT | 0: No WAIT (The period between ninth clock cycle and first clock cycle is not held low.)<br>1: WAIT (The period between ninth clock cycle and first clock cycle is held low.) Low-hold is released by reading ICDRR. | R/W*2 |
| b7 | SMBS | SMBus/I2C Bus Selection | 0: The I2C bus is selected.<br>1: The SMBus is selected. | R/W |

Note 1. Write to the ACKBT bit only while the ACKWP bit is already 1. If it is attempted to write 1 to both the ACKWP and ACKBT bits at the same time, the ACKBT bit will not be set to 1.
Note 2. The WAIT and RDRFS bits are valid only in receive mode (invalid in transmit mode).

### NF[1:0] Bits (Number of Noise Filter Stages Select)

These bits are used to select the number of stages in the digital noise filter.

For details on the digital noise filter function, refer to section 27.6, Digital Noise-Filter Circuits.

Note: • Set the noise range to be filtered out by the noise filter within a range less than the SCL0 line high-level period or low-level period. If the noise range is set to a value of (SCL clock width: high-level period or low-level period, whichever is shorter) - [1.5 internal reference clock (IICϕ) cycles + analog noise filter: 120 ns (reference values)] or more, the SCL clock is regarded as noise by the noise filter function of the RIIC, which may prevent the RIIC from operating normally.

### ACKBR Bit (Receive Acknowledge)

This bit is used to store the acknowledge bit information received from the receive device in transmit mode.
[Setting condition]

- When 1 is received as the acknowledge bit with the TRS bit in ICCR2 set to 1

[Clearing conditions]

- When 0 is received as the acknowledge bit with the TRS bit in ICCR2 set to 1
- When 1 is written to the IICRST bit in ICCR1 while the ICE bit in ICCR1 is 0 (RIIC reset)

### ACKBT Bit (Transmit Acknowledge)

This bit is used to set the bit to be sent at the acknowledge timing in receive mode.
[Setting condition]

- When 1 is written to this bit with the ACKWP bit set to 1

[Clearing conditions]

- When 0 is written to this bit with the ACKWP bit set to 1
- When stop condition issuance is detected (when a stop condition is detected with the SP bit in ICCR2 set to 1)
- When 1 is written to the IICRST bit in ICCR1 while the ICE bit in ICCR1 is 0 (RIIC reset)

### ACKWP Bit (ACKBT Write Protect)

This bit is used to control the modification of the ACKBT bit.

### RDRFS Bit (RDRF Flag Set Timing Selection)

This bit is used to select the RDRF flag set timing in receive mode and also to select whether to hold the SCL0 line low at the falling edge of the eighth SCL clock cycle.
When the RDRFS bit is 0, the SCL0 line is not held low at the falling edge of the eighth SCL clock cycle, and the RDRF flag is set to 1 at the rising edge of the ninth SCL clock cycle.
When the RDRFS bit is 1, the RDRF flag is set to 1 at the rising edge of the eighth SCL clock cycle and the SCL0 line is held low at the falling edge of the eighth SCL clock cycle. The low-hold of the SCL0 line is released by writing a value to the ACKBT bit.
After data is received with this setting, the SCL0 line is automatically held low before the acknowledge bit is sent. This enables processing to send ACK (ACKBT = 0) or NACK (ACKBT = 1) according to receive data.

### WAIT Bit (WAIT)

This bit is used to control whether to hold the period between the ninth SCL clock cycle and the first SCL clock cycle low until the receive data buffer (ICDRR) is completely read each time single-byte data is received in receive mode.
When the WAIT bit is 0, the receive operation is continued without holding the period between the ninth and the first SCL clock cycle low. When both the RDRFS and WAIT bits are 0, continuous receive operation is enabled with the double buffer.
When the WAIT bit is 1, the SCL0 line is held low from the falling edge of the ninth clock cycle until the ICDRR value is read each time single-byte data is received. This enables receive operation in byte units.

Note: • When the value of the WAIT bit is to be read, be sure to read the ICDRR beforehand.

### SMBS Bit (SMBus/I2C Bus Selection)

Setting this bit to 1 selects the SMBus and enables the HOAE bit in ICSER.

## 27.2.6 I2C Bus Function Enable Register (ICFER)

Address(es): RIIC0.ICFER 0008 8305h

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | SCLE | NFE | NACKE | SALE | NALE | MALE | TMOE |
| Value after reset: | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b0 | TMOE | Timeout Function Enable | 0: The timeout function is disabled.<br>1: The timeout function is enabled. | R/W |
| b1 | MALE | Master Arbitration-Lost Detection Enable | 0: Master arbitration-lost detection is disabled.<br>(Disables the arbitration-lost detection function and does not clear the MST and TRS bits in ICCR2 automatically when arbitration is lost.)<br>1: Master arbitration-lost detection is enabled.<br>(Enables the arbitration-lost detection function and clears the MST and TRS bits in ICCR2 automatically when arbitration is lost.) | R/W |
| b2 | NALE | NACK Transmission Arbitration-Lost Detection Enable | 0: NACK transmission arbitration-lost detection is disabled.<br>1: NACK transmission arbitration-lost detection is enabled. | R/W |
| b3 | SALE | Slave Arbitration-Lost Detection Enable | 0: Slave arbitration-lost detection is disabled.<br>1: Slave arbitration-lost detection is enabled. | R/W |
| b4 | NACKE | NACK Reception Transfer Suspension Enable | 0: Transfer operation is not suspended during NACK reception (transfer suspension disabled).<br>1: Transfer operation is suspended during NACK reception (transfer suspension enabled). | R/W |
| b5 | NFE | Digital Noise Filter Circuit Enable | 0: No digital noise filter circuit is used.<br>1: A digital noise filter circuit is used. | R/W |
| b6 | SCLE | SCL Synchronous Circuit Enable | 0: No SCL synchronous circuit is used.<br>1: An SCL synchronous circuit is used. | R/W |
| b7 | — | Reserved | This bit is read as 0. The write value should be 0. | R/W |

### TMOE Bit (Timeout Function Enable)

This bit is used to enable or disable the timeout function.

For details on the timeout function, refer to section 27.11.1, Timeout Function.

### MALE Bit (Master Arbitration-Lost Detection Enable)

This bit is used to specify whether to use the arbitration-lost detection function in master mode. Normally, set this bit to 1.

### NALE Bit (NACK Transmission Arbitration-Lost Detection Enable)

This bit is used to specify whether to cause arbitration to be lost when ACK is detected during transmission of NACK in receive mode (such as when slaves with the same address exist on the bus or when two or more masters select the same slave device simultaneously with different number of receive bytes).

### SALE Bit (Slave Arbitration-Lost Detection Enable)

This bit is used to specify whether to cause arbitration to be lost when a value different from the value being transmitted is detected on the bus in slave transmit mode (such as when slaves with the same address exist on the bus or when a mismatch with the transmit data occurs due to noise).

### NACKE Bit (NACK Reception Transfer Suspension Enable)

This bit is used to specify whether to continue or discontinue the transfer operation when NACK is received from the slave device in transmit mode. Normally, set this bit to 1.
When NACK is received with the NACKE bit set to 1, the next transfer operation is suspended.
When the NACKE bit is 0, the next transfer operation is continued regardless of the received acknowledge content.
For details on the NACK reception transfer suspension function, refer to section 27.8.2, NACK Reception Transfer Suspension Function.

### SCLE Bit (SCL Synchronous Circuit Enable)

This bit is used to specify whether to synchronize the SCL clock with the SCL input clock. Normally, set this bit to 1.
When the SCLE bit is cleared to 0 (no SCL synchronous circuit used), the RIIC does not synchronize the SCL clock with the SCL input clock. In this setting, the RIIC outputs the SCL clock with the transfer rate set in ICBRH and ICBRL regardless of the SCL0 line state. For this reason, if the bus load of the I2C bus line is much larger than the specification value or if the SCL clock output overlaps in multiple masters, the short-cycle SCL clock that does not meet the specification may be output. When no SCL synchronous circuit is used, it also affects the issuance of a start condition, restart condition, and stop condition, and the continuous output of extra SCL clock cycles.
This bit must not be cleared to 0 except for checking the output of the set transfer rate.

## 27.2.7 I2C Bus Status Enable Register (ICSER)

Address(es): RIIC0.ICSER 0008 8306h

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | HOAE | — | DIDE | — | GCAE | SAR2E | SAR1E | SAR0E |
| Value after reset: | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b0 | SAR0E | Slave Address Register 0 Enable | 0: Slave address in SARL0 and SARU0 is disabled.<br>1: Slave address in SARL0 and SARU0 is enabled. | R/W |
| b1 | SAR1E | Slave Address Register 1 Enable | 0: Slave address in SARL1 and SARU1 is disabled.<br>1: Slave address in SARL1 and SARU1 is enabled. | R/W |
| b2 | SAR2E | Slave Address Register 2 Enable | 0: Slave address in SARL2 and SARU2 is disabled.<br>1: Slave address in SARL2 and SARU2 is enabled. | R/W |
| b3 | GCAE | General Call Address Enable | 0: General call address detection is disabled.<br>1: General call address detection is enabled. | R/W |
| b4 | — | Reserved | This bit is read as 0. The write value should be 0. | R/W |
| b5 | DIDE | Device-ID Address Detection Enable | 0: Device-ID address detection is disabled.<br>1: Device-ID address detection is enabled. | R/W |
| b6 | — | Reserved | This bit is read as 0. The write value should be 0. | R/W |
| b7 | HOAE | Host Address Enable | 0: Host address detection is disabled.<br>1: Host address detection is enabled. | R/W |

### SARyE Bit (Slave Address Register y Enable) (y = 0 to 2)

This bit is used to enable or disable the received slave address and the slave address set in SARLy and SARUy.

When this bit is set to 1, the slave address set in SARLy and SARUy is enabled and is compared with the received slave address.

When this bit is cleared to 0, the slave address set in SARLy and SARUy is disabled and is ignored even if it matches the received slave address.

### GCAE Bit (General Call Address Enable)

This bit is used to specify whether to ignore the general call address (0000 000b + 0 [W]: All 0) when it is received.

When this bit is set to 1, if the received slave address matches the general call address, the RIIC recognizes the received slave address as the general call address independently of the slave addresses set in SARLy and SARUy (y = 0 to 2) and performs data receive operation.

When this bit is cleared to 0, the received slave address is ignored even if it matches the general call address.

### DIDE Bit (Device-ID Address Detection Enable)

This bit is used to specify whether to recognize and execute the Device-ID address when a device ID (1111 100b) is received in the first frame after a start condition or restart condition is detected.

When this bit is set to 1, if the received first frame matches the device ID, the RIIC recognizes that the Device-ID address has been received. When the following R/W# bit is 0 [W], the RIIC recognizes the second and the following frames as slave addresses and continues the receive operation.

When this bit is cleared to 0, the RIIC ignores the received first frame even if it matches the device ID address and recognizes the first frame as a normal slave address.

For details on the device-ID address detection, refer to section 27.7.3, Device-ID Address Detection.

**HOAE Bit (Host Address Enable)**

This bit is used to specify whether to ignore received host address (0001 000b) when the SMBS bit in ICMR3 is 1.
When this bit is set to 1 while the SMBS bit in ICMR3 is 1, if the received slave address matches the host address, the RIIC recognizes the received slave address as the host address independently of the slave addresses set in SARLy and SARUy (y = 0 to 2) and performs the receive operation.
When the SMBS bit in ICMR3 or the HOAE bit is cleared to 0, the received slave address is ignored even if it matches the host address.

## 27.2.8 I2C Bus Interrupt Enable Register (ICIER)

Address(es): RIIC0.ICIER 0008 8307h

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | TIE | TEIE | RIE | NAKIE | SPIE | STIE | ALIE | TMOIE |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b0 | TMOIE | Timeout Interrupt Request Enable | 0: Timeout interrupt request (TMOI) is disabled.<br>1: Timeout interrupt request (TMOI) is enabled. | R/W |
| b1 | ALIE | Arbitration-Lost Interrupt Request Enable | 0: Arbitration-lost interrupt request (ALI) is disabled.<br>1: Arbitration-lost interrupt request (ALI) is enabled. | R/W |
| b2 | STIE | Start Condition Detection Interrupt Request Enable | 0: Start condition detection interrupt request (STI) is disabled.<br>1: Start condition detection interrupt request (STI) is enabled. | R/W |
| b3 | SPIE | Stop Condition Detection Interrupt Request Enable | 0: Stop condition detection interrupt request (SPI) is disabled.<br>1: Stop condition detection interrupt request (SPI) is enabled. | R/W |
| b4 | NAKIE | NACK Reception Interrupt Request Enable | 0: NACK reception interrupt request (NAKI) is disabled.<br>1: NACK reception interrupt request (NAKI) is enabled. | R/W |
| b5 | RIE | Receive Data Full Interrupt Request Enable | 0: Receive data full interrupt request (RXI) is disabled.<br>1: Receive data full interrupt request (RXI) is enabled. | R/W |
| b6 | TEIE | Transmit End Interrupt Request Enable | 0: Transmit end interrupt request (TEI) is disabled.<br>1: Transmit end interrupt request (TEI) is enabled. | R/W |
| b7 | TIE | Transmit Data Empty Interrupt Request Enable | 0: Transmit data empty interrupt request (TXI) is disabled.<br>1: Transmit data empty interrupt request (TXI) is enabled. | R/W |

### TMOIE Bit (Timeout Interrupt Request Enable)

This bit is used to enable or disable timeout interrupt requests (TMOI) when the TMOF flag in ICSR2 is set to 1. A TMOI interrupt request is canceled by clearing the TMOF flag or the TMOIE bit to 0.

### ALIE Bit (Arbitration-Lost Interrupt Request Enable)

This bit is used to enable or disable arbitration-lost interrupt requests (ALI) when the AL flag in ICSR2 is set to 1. An ALI interrupt request is canceled by clearing the AL flag or the ALIE bit to 0.

### STIE Bit (Start Condition Detection Interrupt Request Enable)

This bit is used to enable or disable start condition detection interrupt requests (STI) when the START flag in ICSR2 is set to 1. An STI interrupt request is canceled by clearing the START flag or the STIE bit to 0.

### SPIE Bit (Stop Condition Detection Interrupt Request Enable)

This bit is used to enable or disable stop condition detection interrupt requests (SPI) when the STOP flag in ICSR2 is set to 1. An SPI interrupt request is canceled by clearing the STOP flag or the SPIE bit to 0.

### NAKIE Bit (NACK Reception Interrupt Request Enable)

This bit is used to enable or disable NACK reception interrupt requests (NAKI) when the NACKF flag in ICSR2 is set to 1. An NAKI interrupt request is canceled by clearing the NACKF flag or the NAKIE bit to 0.

### RIE Bit (Receive Data Full Interrupt Request Enable)

This bit is used to enable or disable receive data full interrupt requests (RXI) when the RDRF flag in ICSR2 is set to 1.

**TEIE Bit (Transmit End Interrupt Request Enable)**

This bit is used to enable or disable transmit end interrupt requests (TEI) when the TEND flag in ICSR2 is set to 1. An TEI interrupt request is canceled by clearing the TEND flag or the TEIE bit to 0.

**TIE Bit (Transmit Data Empty Interrupt Request Enable)**

This bit is used to enable or disable transmit data empty interrupt requests (TXI) when the TDRE flag in ICSR2 is set to 1.

## 27.2.9 I2C Bus Status Register 1 (ICSR1)

Address(es): RIIC0.ICSR1 0008 8308h

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | HOA | — | DID | — | GCA | AAS2 | AAS1 | AAS0 |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b0 | AAS0 | Slave Address 0 Detection Flag | 0: Slave address 0 is not detected.<br>1: Slave address 0 is detected. | R/(W) *1 |
| b1 | AAS1 | Slave Address 1 Detection Flag | 0: Slave address 1 is not detected.<br>1: Slave address 1 is detected. | R/(W) *1 |
| b2 | AAS2 | Slave Address 2 Detection Flag | 0: Slave address 2 is not detected.<br>1: Slave address 2 is detected. | R/(W) *1 |
| b3 | GCA | General Call Address Detection Flag | 0: General call address is not detected.<br>1: General call address is detected. | R/(W) *1 |
| b4 | — | Reserved | This bit is read as 0. The write value should be 0. | R/W |
| b5 | DID | Device-ID Address Detection Flag | 0: Device-ID command is not detected.<br>1: Device-ID command is detected.<br>• This bit is set to 1 when the first frame received immediately after a start condition is detected matches a value of (device ID (1111 100b) + 0[W]). | R/(W) *1 |
| b6 | — | Reserved | This bit is read as 0. The write value should be 0. | R/W |
| b7 | HOA | Host Address Detection Flag | 0: Host address is not detected.<br>1: Host address is detected.<br>• This bit is set to 1 when the received slave address matches the host address (0001 000b). | R/(W) *1 |

Note 1. Only 0 can be written to clear the flag.

**AASy Flag (Slave Address y Detection Flag) (y = 0 to 2)**

[Setting conditions]

For 7-bit address format: SARUy.FS = 0

- When the received slave address matches the SVA[6:0] value in SARLy with the SARyE bit in ICSER set to 1 (slave address y detection enabled)
  This flag is set to 1 at the rising edge of the ninth SCL clock cycle in the frame.

For 10-bit address format: SARUy.FS = 1

- When the received slave address matches a value of (11110b + SVA[1:0] in SARUy) and the following address matches the SARLy value with the SARyE bit in ICSER set to 1 (slave address y detection enabled)
  This flag is set to 1 at the rising edge of the ninth SCL clock cycle in the frame.

[Clearing conditions]

- When 0 is written to the AASy bit after reading AASy = 1
- When a stop condition is detected
- When 1 is written to the IICRST bit in ICCR1 to apply an RIIC reset or an internal reset

For 7-bit address format: SARUy.FS = 0

- When the received slave address does not match the SVA[6:0] value in SARLy with the SARyE bit in ICSER set to 1 (slave address y detection enabled)
  This flag is cleared to 0 at the rising edge of the ninth SCL clock cycle in the frame.

For 10-bit address format: SARUy.FS = 1

- When the received slave address does not match a value of (11110b + SVA[1:0] in SARUy) with the SARyE bit in ICSER set to 1 (slave address y detection enabled)
  This flag is cleared to 0 at the rising edge of the ninth SCL clock cycle in the frame.
- When the received slave address matches a value of (11110b + SVA[1:0] in SARUy) and the following address does not match the SARLy value with the SARyE bit in ICSER set to 1 (slave address y detection enabled)
  This flag is cleared to 0 at the rising edge of the ninth SCL clock cycle in the frame.

#### GCA Flag (General Call Address Detection Flag)

[Setting condition]

- When the received slave address matches the general call address (0000 000b + 0 [W]) with the GCAE bit in ICSER set to 1 (general call address detection is enabled)
  This flag is set to 1 at the rising edge of the ninth SCL clock cycle in the frame.

[Clearing conditions]

- When 0 is written to the GCA bit after reading GCA = 1
- When a stop condition is detected
- When the received slave address does not match the general call address (0000 000b + 0 [W]) with the GCAE bit in ICSER set to 1 (general call address detection is enabled)
  This flag is cleared to 0 at the rising edge of the ninth SCL clock cycle in the frame.
- When 1 is written to the IICRST bit in ICCR1 to apply an RIIC reset or an internal reset

**DID Flag (Device-ID Address Detection Flag)**

[Setting condition]

- When the first frame received immediately after a start condition or restart condition is detected matches a value of (device ID (1111 100b) + 0 [W]) with the DIDE bit in ICSER set to 1 (Device-ID address detection is enabled)
  This flag is set to 1 at the rising edge of the ninth SCL clock cycle in the frame.

[Clearing conditions]

- When 0 is written to the DID bit after reading DID = 1
- When a stop condition is detected
- When the first frame received immediately after a start condition or restart condition is detected does not match a value of (device ID (1111 100b)) with the DIDE bit in ICSER set to 1 (Device-ID address detection is enabled)
  This flag is cleared to 0 at the rising edge of the ninth SCL clock cycle in the frame.
- When the first frame received immediately after a start condition or restart condition is detected matches a value of (device ID (1111 100b) + 0 [W]) and the second frame does not match any of slave addresses 0 to 2 with the DIDE bit in ICSER set to 1 (Device-ID address detection is enabled)
  This flag is cleared to 0 at the rising edge of the ninth SCL clock cycle in the frame.
- When 1 is written to the IICRST bit in ICCR1 to apply an RIIC reset or an internal reset

**HOA Flag (Host Address Detection Flag)**

[Setting condition]

- When the received slave address matches the host address (0001 000b) with the HOAE bit in ICSER set to 1 (host address detection is enabled)
  This flag is set to 1 at the rising edge of the ninth SCL clock cycle in the frame.

[Clearing conditions]

- When 0 is written to the HOA bit after reading HOA = 1
- When a stop condition is detected
- When 0 is written to the SMBS bit in ICMR3 or the HOAE bit in ICSER
- When the received slave address does not match the host address (0001 000b) with the HOAE bit in ICSER set to 1 (host address detection is enabled)
  This flag is cleared to 0 at the rising edge of the ninth SCL clock cycle in the frame.
- When 1 is written to the IICRST bit in ICCR1 to apply an RIIC reset or an internal reset

## 27.2.10 I2C Bus Status Register 2 (ICSR2)

Address(es): RIIC0.ICSR2 0008 8309h

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | TDRE | TEND | RDRF | NACKF | STOP | START | AL | TMOF |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b0 | TMOF | Timeout Detection Flag | 0: Timeout is not detected.<br>1: Timeout is detected. | R/(W) *1 |
| b1 | AL | Arbitration-Lost Flag | 0: Arbitration is not lost.<br>1: Arbitration is lost. | R/(W) *1 |
| b2 | START | Start Condition Detection Flag | 0: Start condition is not detected.<br>1: Start condition is detected. | R/(W) *1 |
| b3 | STOP | Stop Condition Detection Flag | 0: Stop condition is not detected.<br>1: Stop condition is detected. | R/(W) *1 |
| b4 | NACKF | NACK Detection Flag | 0: NACK is not detected.<br>1: NACK is detected. | R/(W) *1 |
| b5 | RDRF | Receive Data Full Flag | 0: ICDRR contains no receive data.<br>1: ICDRR contains receive data. | R/(W) *1 |
| b6 | TEND | Transmit End Flag | 0: Data is being transmitted.<br>1: Data has been transmitted. | R/(W) *1 |
| b7 | TDRE | Transmit Data Empty Flag | 0: ICDRT contains transmit data.<br>1: ICDRT contains no transmit data. | R |

Note 1. Only 0 can be written to clear the flag.

### TMOF Flag (Timeout Detection Flag)

This flag is set to 1 when the RIIC recognizes timeout after the SCL0 line state remains unchanged for a certain period.
[Setting condition]

- When the SCL0 line state remains unchanged for the period specified by the ICMR2.TMOH, TMOL, and TMOS bits while the ICFER.TMOE bit is 1 (the timeout function is enabled) in master mode or in slave mode and the received slave address matches.

[Clearing conditions]

- When 0 is written to the TMOF bit after reading TMOF = 1
- When 1 is written to the IICRST bit in ICCR1 to apply an RIIC reset or an internal reset

### AL Flag (Arbitration-Lost Flag)

This flag shows that bus mastership has been lost (loss in arbitration) due to a bus conflict or some other reason when a start condition is issued or an address and data are transmitted. The RIIC monitors the level on the SDA0 line during transmission and, if the level on the line does not match the value of the bit being output, sets the value of the AL flag to 1 to indicate that the bus is occupied by another device.
The RIIC can also set the flag to indicate the detection of loss of arbitration during NACK transmission in master mode or during data transmission in slave mode.

[Setting conditions]

When master arbitration-lost detection is enabled: ICFER.MALE = 1

- When the internal SDA output state does not match the SDA0 line level at the rising edge of SCL clock except for the ACK period during data (including slave address) transmission in master transmit mode (when the SDA0 line is driven low while the internal SDA output is at a high level (the SDA0 pin is in the high-impedance state))
- When a start condition is detected while the ST bit in ICCR2 is 1 (start condition issuance request) or the internal SDA output state does not match the SDA0 line level
- When the ST bit in ICCR2 is set to 1 (start condition issuance request) with the BBSY flag in ICCR2 set to 1.

When NACK arbitration-lost detection is enabled: ICFER.NALE = 1

- When the internal SDA output state does not match the SDA0 line level at the rising edge of SCL clock in the ACK period during NACK transmission in receive mode

When slave arbitration-lost detection is enabled: ICFER.SALE = 1

- When the internal SDA output state does not match the SDA0 line level at the rising edge of SCL clock except for the ACK period during data transmission in slave transmit mode

[Clearing conditions]

- When 0 is written to the AL flag after reading AL = 1
- When 1 is written to the IICRST bit in ICCR1 to apply an RIIC reset or an internal reset

**Table 27.4 Relationship between Arbitration-Lost Generation Sources and Arbitration-Lost Enable Functions**

| ICFER | | | ICSR2 | | |
|---|---|---|---|---|---|
| MALE | NALE | SALE | AL | Error | Arbitration-Lost Generation Source |
| 1 | × | × | 1 | Start condition issuance error | When internal SDA output state does not match SDA0 line level when a start condition is detected while the ST bit in ICCR2 is 1 |
| | | | | | When ST in ICCR2 is set to 1 with BBSY in ICCR2 set to 1 |
| | | | 1 | Transmit data mismatch | When transmit data (including slave address) does not match the bus state in master transmit mode |
| × | 1 | × | 1 | NACK transmission mismatch | When ACK is detected during transmission of NACK in master receive mode or slave receive mode |
| × | × | 1 | 1 | Transmit data mismatch | When transmit data does not match the bus state in slave transmit mode |

×: Don't care

### START Flag (Start Condition Detection Flag)

[Setting condition]

- When a start condition (or a restart condition) is detected

[Clearing conditions]

- When 0 is written to the START bit after reading START = 1
- When a stop condition is detected
- When 1 is written to the IICRST bit in ICCR1 to apply an RIIC reset or an internal reset

### STOP Flag (Stop Condition Detection Flag)

[Setting condition]

- When a stop condition is detected

[Clearing conditions]

- When 0 is written to the STOP bit after reading STOP = 1
- When 1 is written to the IICRST bit in ICCR1 to apply an RIIC reset or an internal reset

### NACKF Flag (NACK Detection Flag)

[Setting condition]

- When acknowledge is not received (NACK is received) from the receive device in transmit mode with the NACKE bit in ICFER set to 1 (transfer suspension enabled)

[Clearing conditions]

- When 0 is written to the NACKF bit after reading NACKF = 1
- When 1 is written to the IICRST bit in ICCR1 to apply an RIIC reset or an internal reset

Note: • When the NACKF flag is set to 1, the RIIC suspends data transmission/reception. Writing to ICDRT in transmit mode or reading from ICDRR in receive mode with the NACKF flag set to 1 does not enable data transmit/receive operation. To restart data transmission/reception, clear the NACKF flag to 0.

### RDRF Flag (Receive Data Full Flag)

[Setting conditions]

- When receive data has been transferred from ICDRS to ICDRR
  This flag is set to 1 at the rising edge of the eighth or ninth SCL clock cycle (selected by the RDRFS bit in ICMR3)
- When the received slave address matches after a start condition (or a restart condition) is detected with the TRS bit in ICCR2 cleared to 0

[Clearing conditions]

- When 0 is written to the RDRF bit after reading RDRF = 1
- When data is read from ICDRR
- When 1 is written to the IICRST bit in ICCR1 to apply an RIIC reset or an internal reset

### TEND Flag (Transmit End Flag)

[Setting condition]

- At the rising edge of the ninth SCL clock cycle while the TDRE flag is 1

[Clearing conditions]

- When 0 is written to the TEND bit after reading TEND = 1
- When data is written to ICDRT
- When a stop condition is detected
- When 1 is written to the IICRST bit in ICCR1 to apply an RIIC reset or an internal reset

**TDRE Flag (Transmit Data Empty Flag)**

[Setting conditions]

- When data has been transferred from ICDRT to ICDRS and ICDRT becomes empty
- When the TRS bit in ICCR2 is set to 1
- When the received slave address matches while the TRS bit is 1

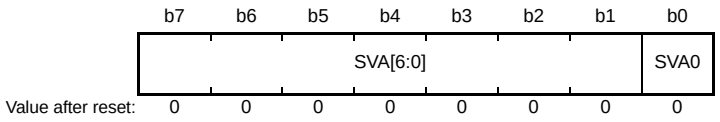
[Clearing conditions]

- When data is written to ICDRT
- When the TRS bit in ICCR2 is cleared to 0
- When 1 is written to the IICRST bit in ICCR1 to apply an RIIC reset or an internal reset

Note: • When the NACKF flag is set to 1 while the NACKE bit in ICFER is 1, the RIIC suspends data transmission/reception. Here, if the TDRE flag is 0 (next transmit data has been written), data is transferred to the ICDRS register and the ICDRT register becomes empty at the rising edge of the ninth clock cycle, but the TDRE flag is not set to 1.

## 27.2.11 Slave Address Register Ly (SARLy) (y = 0 to 2)

Address(es): RIIC0.SARL0 0008 830Ah, RIIC0.SARL1 0008 830Ch, RIIC0.SARL2 0008 830Eh




| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b0 | SVA0 | 10-Bit Address LSB | A slave address is set. | R/W |
| b7 to b1 | SVA[6:0] | 7-Bit Address/10-Bit Address Lower Bits | A slave address is set. | R/W |

### SVA0 Bit (10-Bit Address LSB)

When the 10-bit address format is selected (SARUy.FS = 1), this bit functions as the LSB of a 10-bit address and forms the lower 8 bits of a 10-bit address in combination with the SVA[6:0] bits.
When the SARyE bit in ICSER is set to 1 (SARLy and SARUy enabled) and the SARUy.FS bit is 1, this bit is valid. While the SARUy.FS bit or SARyE bit is 0, the setting of this bit is ignored.

### SVA[6:0] Bits (7-Bit Address/10-Bit Address Lower Bits)

When the 7-bit address format is selected (SARUy.FS = 0), these bits function as a 7-bit address. When the 10-bit address format is selected (SARUy.FS = 1), these bits function as the lower 8 bits of a 10-bit address in combination with the SVA0 bit.
While the SARyE bit in ICSER is 0, the setting of these bits is ignored.

## 27.2.12 Slave Address Register Uy (SARUy) (y = 0 to 2)

Address(es): RIIC0.SARU0 0008 830Bh, RIIC0.SARU1 0008 830Dh, RIIC0.SARU2 0008 830Fh

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | SVA[1:0] | | FS |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b0 | FS | 7-Bit/10-Bit Address Format Selection | 0: The 7-bit address format is selected.<br>1: The 10-bit address format is selected. | R/W |
| b2, b1 | SVA[1:0] | 10-Bit Address Upper Bits | A slave address is set. | R/W |
| b7 to b3 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |

### FS Bit (7-Bit/10-Bit Address Format Selection)

This bit is used to select 7-bit address or 10-bit address for slave address y (in SARLy and SARUy).
When the SARyE bit in ICSER is set to 1 (SARLy and SARUy enabled) and the SARUy.FS bit is 0, the 7-bit address format is selected for slave address y, the SVA[6:0] setting in SARLy is valid, and the settings of the SVA[1:0] bits and the SVA0 bit in SARLy are ignored.
When the SARyE bit in ICSER is set to 1 (SARLy and SARUy enabled) and the SARUy.FS bit is 1, the 10-bit address format is selected for slave address y and the settings of the SVA[1:0] bits and SARLy are valid.
While the SARyE bit in ICSER is 0 (SARLy and SARUy disabled), the setting of the SARUy.FS bit is invalid.

### SVA[1:0] Bits (10-Bit Address Upper Bits)

When the 10-bit address format is selected (FS = 1), these bits function as the upper 2 bits of a 10-bit address.
When the SARyE bit in ICSER is set to 1 (SARLy and SARUy enabled) and the SARUy.FS bit is 1, these bits are valid.
While the SARUy.FS bit or SARyE bit is 0, the setting of these bits is ignored.

## 27.2.13 I2C Bus Bit Rate Low-Level Register (ICBRL)

Address(es): RIIC0.ICBRL 0008 8310h

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | — | BRL[4:0] | | | | |
| Value after reset: | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

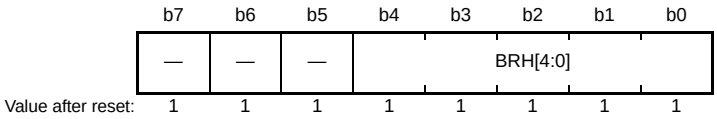
| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b4 to b0 | BRL[4:0] | Bit Rate Low-Level Period | Low-level period of SCL clock | R/W |
| b7 to b5 | — | Reserved | These bits are read as 1. The write value should be 1. | R/W |

ICBRL is a 5-bit register to set the low-level period of SCL clock.
It also works to generate the data setup time for automatic SCL low-hold operation (refer to section 27.8, Automatic Low-Hold Function for SCL); when the RIIC is used only in slave mode, this register needs to be set to a value longer than the data setup time*1.
ICBRL counts the low-level period with the internal reference clock source (IICϕ) specified by the CKS[2:0] bits in ICMR1.
If the digital noise filter is enabled (the NFE bit in ICFER is 1), set the ICBRL register to a value at least one greater than the number of stages in the noise filter. Regarding the number of stages in the noise filter, see the description of the ICMR3.NF[1:0] bits.

Note 1. Data setup time (tSU: DAT)
250 ns (up to 100 kbps: standard mode [Sm])
100 ns (up to 400 kbps: fast mode [Fm])

## 27.2.14 I2C Bus Bit Rate High-Level Register (ICBRH)

Address(es): RIIC0.ICBRH 0008 8311h




| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b4 to b0 | BRH[4:0] | Bit Rate High-Level Period | High-level period of SCL clock | R/W |
| b7 to b5 | — | Reserved | These bits are read as 1. The write value should be 1. | R/W |

ICBRH is a 5-bit register to set the high-level period of SCL clock. ICBRH is valid in master mode. If the RIIC is used only in slave mode, this register need not to set the high-level period.
ICBRH counts the high-level period with the internal reference clock source (IICϕ) specified by the CKS[2:0] bits in ICMR1.
If the digital noise filter is enabled (the NFE bit in ICFER is 1), set the ICBRH register to a value at least one greater than the number of stages in the noise filter. Regarding the number of stages in the noise filter, see the description of the ICMR3.NF[1:0] bits.

The I2C transfer rate and the SCL clock duty are calculated using the following expression.
Transfer rate = 1 / {[(ICBRH + 1) + (ICBRL + 1)] / IICϕ*1 + SCL0 line rising time [tr] + SCL0 line falling time [tf]}
Duty cycle = {SCL0 line rising time [tr]*2 + (ICBRH + 1) / IICϕ} / {SCL0 line falling time [tf]*2 + (ICBRL + 1) / IICϕ}

Note 1. IICϕ = PCLK × Division ratio
Note 2. The SCL0 line rising time [tr] and SCL0 line falling time [tf] depend on the total bus line capacitance [Cb] and the pull-up resistor [Rp]. For details, see the I2C bus standard from NXP Semiconductors.

Table 27.5 lists examples of ICBRH/ICBRL settings.

**Table 27.5 Examples of ICBRH/ICBRL Settings for Transfer Rate**

| Transfer Rate (kbps) | Operating Frequency PCLK (MHz) | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | 8 | | | 10 | | | 12.5 | | |
| | CKS[2:0] | ICBRH | ICBRL | CKS[2:0] | ICBRH | ICBRL | CKS[2:0] | ICBRH | ICBRL |
| 10 | 100b | 22 (F6h) | 25 (F9h) | 101b | 13 (EDh) | 15 (EFh) | 101b | 16 (F0h) | 20 (F4h) |
| 50 | 010b | 16 (F0h) | 19 (F3h) | 010b | 21 (F5h) | 24 (F8h) | 011b | 12 (ECh) | 15 (EFh) |
| 100 | 001b | 15 (EFh) | 18 (F2h) | 001b | 19 (F3h) | 23 (F7h) | 001b | 24 (F8h) | 29 (FDh) |
| 400 | 000b | 4 (E4h) | 10 (EAh) | 000b | 5 (E5h) | 12 (ECh) | 000b | 7 (E7h) | 16 (F0h) |

| Transfer Rate (kbps) | Operating Frequency PCLK (MHz) | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | 16 | | | 20 | | | 25 | | |
| | CKS[2:0] | ICBRH | ICBRL | CKS[2:0] | ICBRH | ICBRL | CKS[2:0] | ICBRH | ICBRL |
| 10 | 101b | 22 (F6h) | 25 (F9h) | 110b | 13 (EDh) | 15 (EFh) | 110b | 16 (F0h) | 20 (F4h) |
| 50 | 011b | 16 (F0h) | 19 (F3h) | 011b | 21 (F5h) | 24 (F8h) | 100b | 12 (ECh) | 15 (EFh) |
| 100 | 010b | 15 (EFh) | 18 (F2h) | 010b | 19 (F3h) | 23 (F7h) | 010b | 24 (F8h) | 29 (FDh) |
| 400 | 000b | 9 (E9h) | 20 (F4h) | 000b | 11 (EBh) | 25 (F9h) | 001b | 7 (E7h) | 16 (F0h) |

| Transfer Rate (kbps) | Operating Frequency PCLK (MHz) | | |
|---|---|---|---|
| | 30 | | |
| | CKS[2:0] | ICBRH | ICBRL |
| 10 | 110b | 20 (F4h) | 24 (F8h) |
| 50 | 100b | 15 (EFh) | 18 (F2h) |
| 100 | 010b | 2 (E2h) | 3 (E3h) |
| 400 | 001b | 8 (E8h) | 19 (F3h) |

Note: • ICBRH/ICBRL settings in these tables are calculated using the following values:
SCL0 line rising time (tr): 100 kbps or less, [Sm]: 1000 ns, 400 kbps or less, [Fm]: 300 ns
SCL0 line falling time (tf): 400 kbps or less, [Sm/Fm]: 300 ns
For the specified values of SCL0 line rising time (tr) and SCL0 line falling time (tf), see the I2C bus standard from NXP Semiconductors.

### 27.2.15 I2C Bus Transmit Data Register (ICDRT)

Address(es): RIIC0.ICDRT 0008 8312h

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| Value after reset: | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

When ICDRT detects a space in the I2C bus shift register (ICDRS), it transfers the transmit data that has been written to ICDRT to ICDRS and starts transmitting data in transmit mode.
The double-buffer structure of ICDRT and ICDRS allows continuous transmit operation if the next transmit data has been written to ICDRT while the ICDRS data is being transmitted.
ICDRT can always be read and written. Write transmit data to ICDRT once when a transmit data empty interrupt (TXI) request is generated.

### 27.2.16 I2C Bus Receive Data Register (ICDRR)

Address(es): RIIC0.ICDRR 0008 8313h

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

When 1 byte of data has been received, the received data is transferred from the I2C bus shift register (ICDRS) to ICDRR to enable the next data to be received.
The double-buffer structure of ICDRS and ICDRR allows continuous receive operation if the received data has been read from ICDRR while ICDRS is receiving data.
ICDRR cannot be written. Read data from ICDRR once when a receive data full interrupt (RXI) request is generated.
If ICDRR receives the next receive data before the current data is read from ICDRR (while the RDRF flag in ICSR2 is 1), the RIIC automatically holds the SCL clock low one cycle before the RDRF flag is set to 1 next.

### 27.2.17 I2C Bus Shift Register (ICDRS)

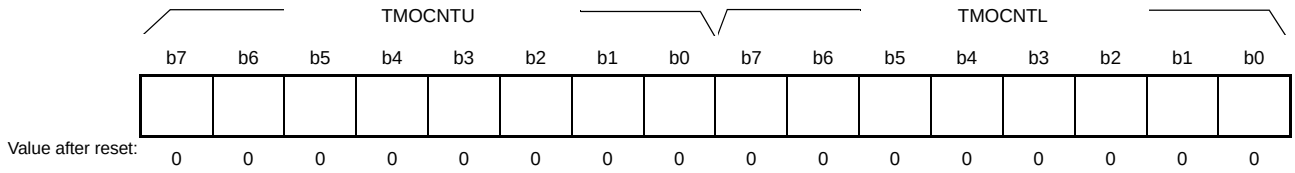
| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| Value after reset: | — | — | — | — | — | — | — | — |

ICDRS is an 8-bit shift register to transmit and receive data.
During transmission, transmit data is transferred from ICDRT to ICDRS and is sent from the SDA0 pin. During reception, data is transferred from ICDRS to ICDRR after 1 byte of data has been received.
ICDRS cannot be accessed directly.

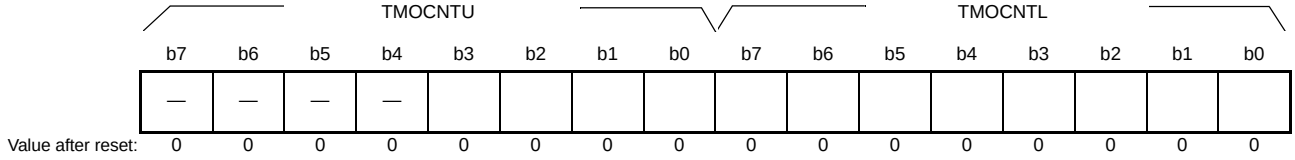
## 27.2.18 Timeout Internal Counter (TMOCNT)

Address(es): RIIC0.TMOCNTL 0008 830Ah, RIIC0.TMOCNTU 0008 830Bh

- TMOS = 0 (Long mode)




- TMOS = 1 (Short mode)




Note: • Same address with ones of the slave address registers, SARL0, SARU0. Care should be taken.

- TMOCNTL register

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b7 to b0 | TMOCNTL | Timeout internal counter | Timeout internal counter low-order | W*1 |

Note 1. Value in timeout internal counter cannot be read. When value is read, the read value is FFh.

- TMOCNTU register

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b7 to b0 | TMOCNTU | Timeout internal counter | Timeout internal counter high-order*1 | W*2 |

Note 1. When TMOS = 1 (short mode), bits b7 to b4 are reserved bits. They are writable. However, the value written is disabled.
Note 2. Value in timeout internal counter cannot be read. When value is read, the read value is FFh.

The timeout internal counter (TMOCNTL/TMOCNTU) is initialized (TMOCNTL = 00h, TMOCNTU = 00h) after a reset, while ICCR1.IICRST = 1 or ICFER.TMOE = 1 and PCLK/1 is selected with ICMR1.CKS[2:0] = 000b setting, and when the counter clear conditions specified by the ICMR2.TMOH and TMOL bits (SCL rising edge/falling edge detection) are met.
Since the timeout internal counter is not automatically initialized when the ICMR1.CKS[2:0] bits are not 000b (PCLK/1), write 00h to the TMOCNTL and TMOCNTU counters as necessary for initialization.
The TMOCNTL and TMOCNTU counters can be accessed as 16-bit registers in 16-bit units.
For 16-bit access, access the address shown in the table below.

**Table 27.6 Register Allocation for 16-Bit Access**

| Address | Upper 8 Bits | Lower 8 Bits |
|---|---|---|
| 0008 830Ah | RIIC0.TMOCNTU | RIIC0.TMOCNTL |

## 27.3 Operation

### 27.3.1 Communication Data Format

The I2C bus format consists of 8-bit data and 1-bit acknowledge. The frame following a start condition or restart condition is an address frame used to specify a slave device with which the master device communicates. The specified slave is valid until a new slave is specified or a stop condition is issued.
Figure 27.3 shows the I2C bus format, and Figure 27.4 shows the I2C bus timing.

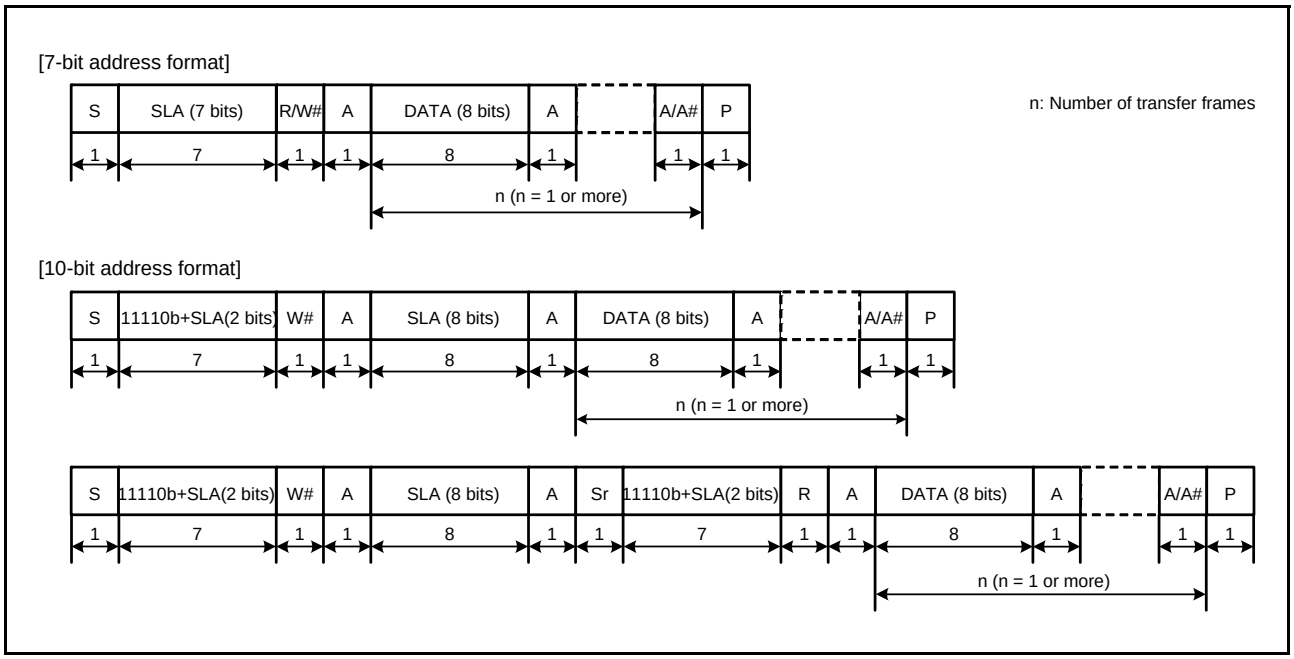



**Figure 27.3 I2C Bus Format**

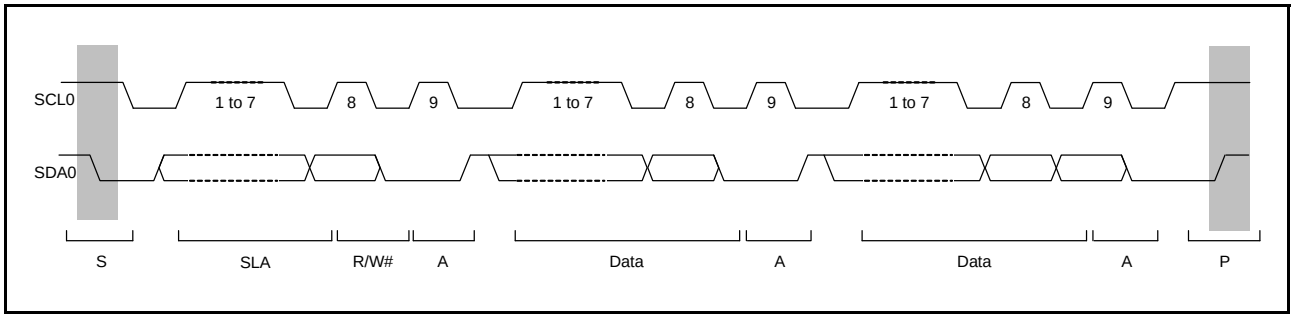



**Figure 27.4 I2C Bus Timing (SLA = 7 Bits)**

S: Start condition. The master device drives the SDA0 line low from high level while the SCL0 line is at a high level.

SLA: Slave address, by which the master device selects a slave device.

R/W#: Indicates the direction of data transfer: from the slave device to the master device when R/W is 1, or from the master device to the slave device when R/W is 0.

A: Acknowledge. The receive device drives the SDA0 line low. (In master transmit mode, the slave device returns acknowledge. In master receive mode, the master device returns acknowledge.)

A#: Not Acknowledge. The receive device drives the SDA0 line high.

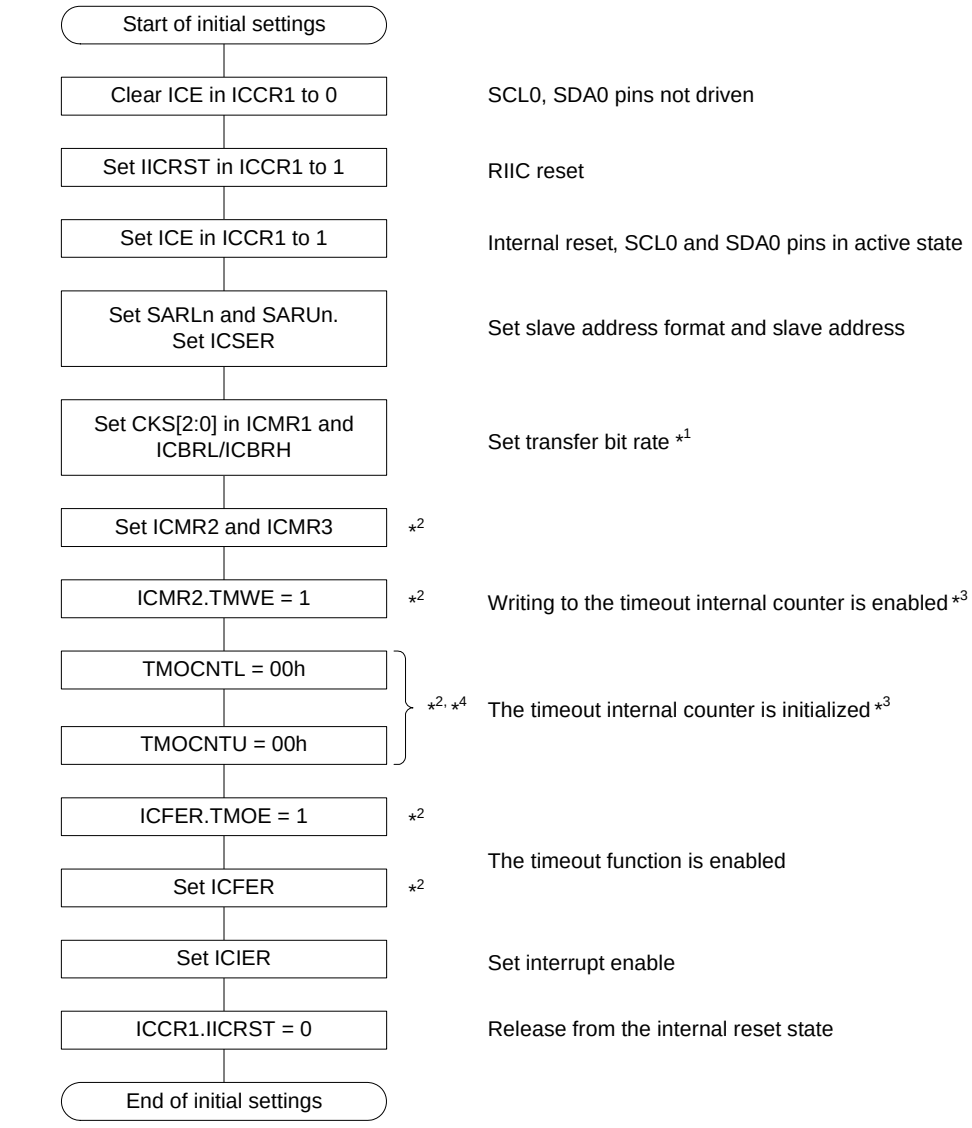
Sr: Restart condition. The master device drives the SDA0 line low from the high level after the setup time has elapsed with the SCL0 line at the high level.

DATA: Transmitted or received data

P: Stop condition. The master device drives the SDA0 line high from low level while the SCL0 line is at a high level.

### 27.3.2 Initial Settings

Before starting data transmission and reception, initialize the RIIC according to the procedure in Figure 27.5.
Set the ICCR1.ICE bit to 1 (internal reset) after setting the ICCR1.IICRST bit to 1 (RIIC reset) with the ICCR1.ICE bit cleared to 0 (SCL0 and SDA0 pins in inactive state). This initializes the various flags and internal state of the ICSR1 register. After that, set registers SARLy, SARUy, ICSER, ICMR1, ICBRH, and ICBRL (y = 0 to 2), and set the other registers as necessary (for initial settings of the RIIC, see Figure 27.5). When the necessary register settings have been completed, clear the ICCR1.IICRST bit to 0 (clears the RIIC reset). This step is not necessary if initialization of the RIIC has already been completed.




[Legend] n = 0 to 2

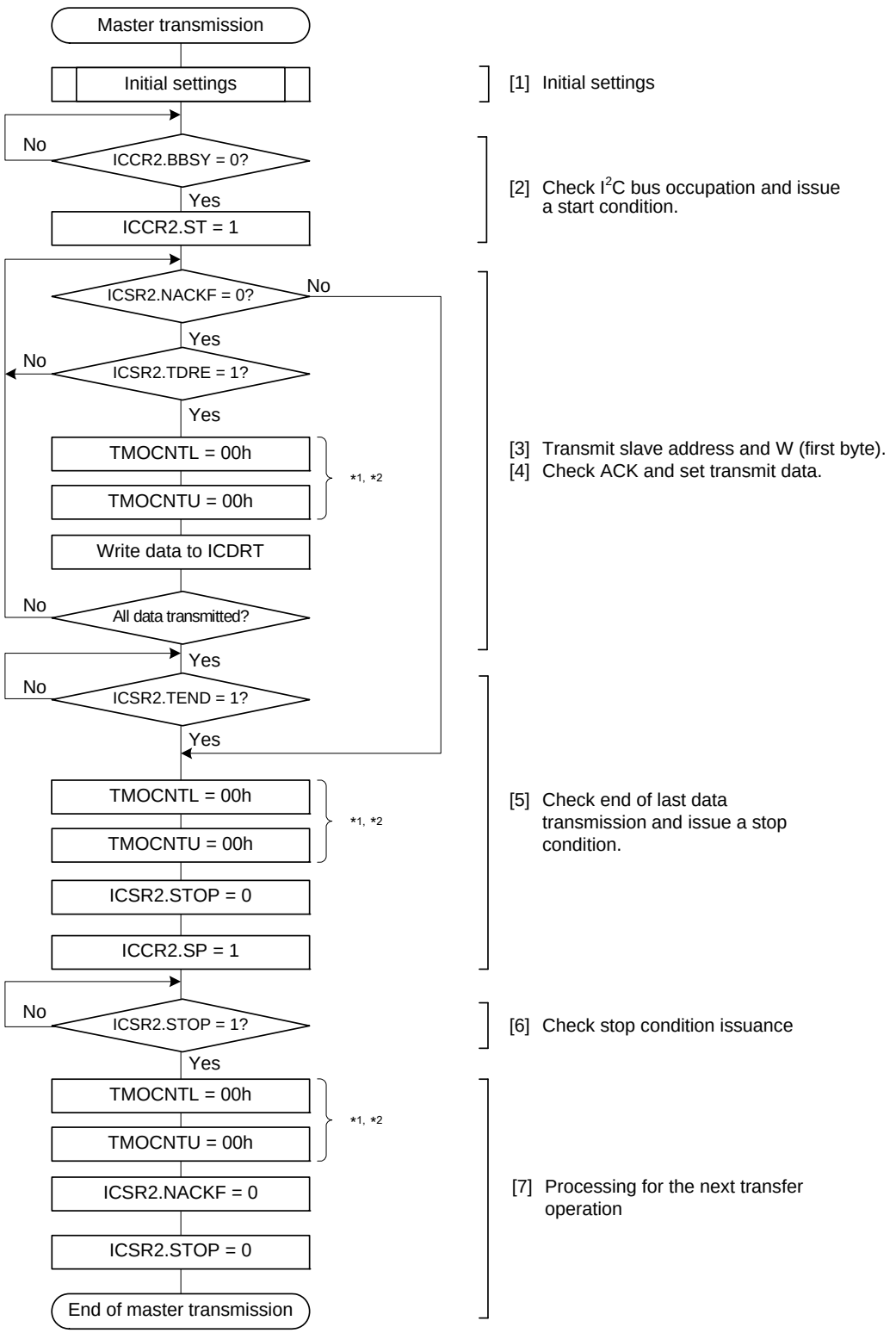
Note 1. When the RIIC is used only in slave mode, set the ICBRL register to a value longer than the data setup time.
Note 2. Set these registers as necessary.
Note 3. These processes are necessary when the timeout function is used and the ICMR1.CKS[2:0] bits are not 000b.
Note 4. Can be accessed as a 16-bit register in words.

**Figure 27.5 Example of RIIC Initialization Flowchart**

### 27.3.3 Master Transmit Operation

In master transmit operation, the RIIC outputs the SCL clock and transmitted data signals as the master device, and the slave device returns acknowledgements. Figure 27.6 shows an example of usage of master transmission and Figure 27.7 to Figure 27.9 show the timing of operations in master transmission.
The following describes the procedure and operations for master transmission.

(1) Initial settings. For details, refer to section 27.3.2, Initial Settings.
(2) Read the BBSY flag in ICCR2 to check that the bus is open, and then set the ST bit in ICCR2 to 1 (start condition issuance request). Upon receiving the request, the RIIC issues a start condition. At the same time, the BBSY flag and the START flag in ICSR2 are automatically set to 1 and the ST bit is automatically cleared to 0. At this time, if the start condition is detected and the internal levels for the SDA output state and the levels on the SDA0 line have matched while the ST bit is 1, the RIIC recognizes that issuing of the start condition as requested by the ST bit has been successfully completed, and the MST and TRS bits in ICCR2 are automatically set to 1, placing the RIIC in master transmit mode. The TDRE flag in ICSR2 is also automatically set to 1 in response to setting of the TRS bit to 1.
(3) Check that the TDRE flag in ICSR2 is 1, and then write the value for transmission (the slave address and the R/W# bit) to ICDRT. Once the data for transmission are written to ICDRT, the TDRE flag is automatically cleared to 0, the data are transferred from ICDRT to ICDRS, and the TDRE flag is again set to 1. After the byte containing the slave address and R/W# bit has been transmitted, the value of the TRS bit is automatically updated to select master transmit or master receive mode in accord with the value of the transmitted R/W# bit. If the value of the R/W# bit was 0, the RIIC continues in master transmit mode.
Since the ICSR2.NACKF flag being 1 at this time indicates that no slave device recognized the address or there was an error in communications, write 1 to the ICCR2.SP bit to issue a stop condition.
For data transmission with an address in the 10-bit format, start by writing 1111 0b, the 2 higher-order bits of the slave address, and W to ICDRT as the first address transmission. Then, as the second address transmission, write the 8 lower-order bits of the slave address to ICDRT.
(4) After confirming that the TDRE flag in ICSR2 is 1, write the data for transmission to the ICDRT register. The RIIC automatically holds the SCL0 line low until the data for transmission are ready or a stop condition is issued.
(5) After all bytes of data for transmission have been written to the ICDRT register, wait until the value of the TEND flag in ICSR2 returns to 1, and then set the SP bit in ICCR2 to 1 (stop condition issuance request). Upon receiving a stop condition issuance request, the RIIC issues the stop condition.
(6) Upon detecting the stop condition, the RIIC automatically clears the MST and TRS bits in ICCR2 to 00b and enters slave receive mode. Furthermore, it automatically clears the TDRE and TEND flags to 0, and sets the STOP flag in ICSR2 to 1.
(7) After checking that the ICSR2.STOP flag is 1, clear the ICSR2.NACKF and STOP flags to 0 for the next transfer operation.




Note 1. These settings are required only when timeout detection is enabled.
Note 2. Word access can be also performed as a 16-bit register.

**Figure 27.6 Example of Master Transmission Flowchart**

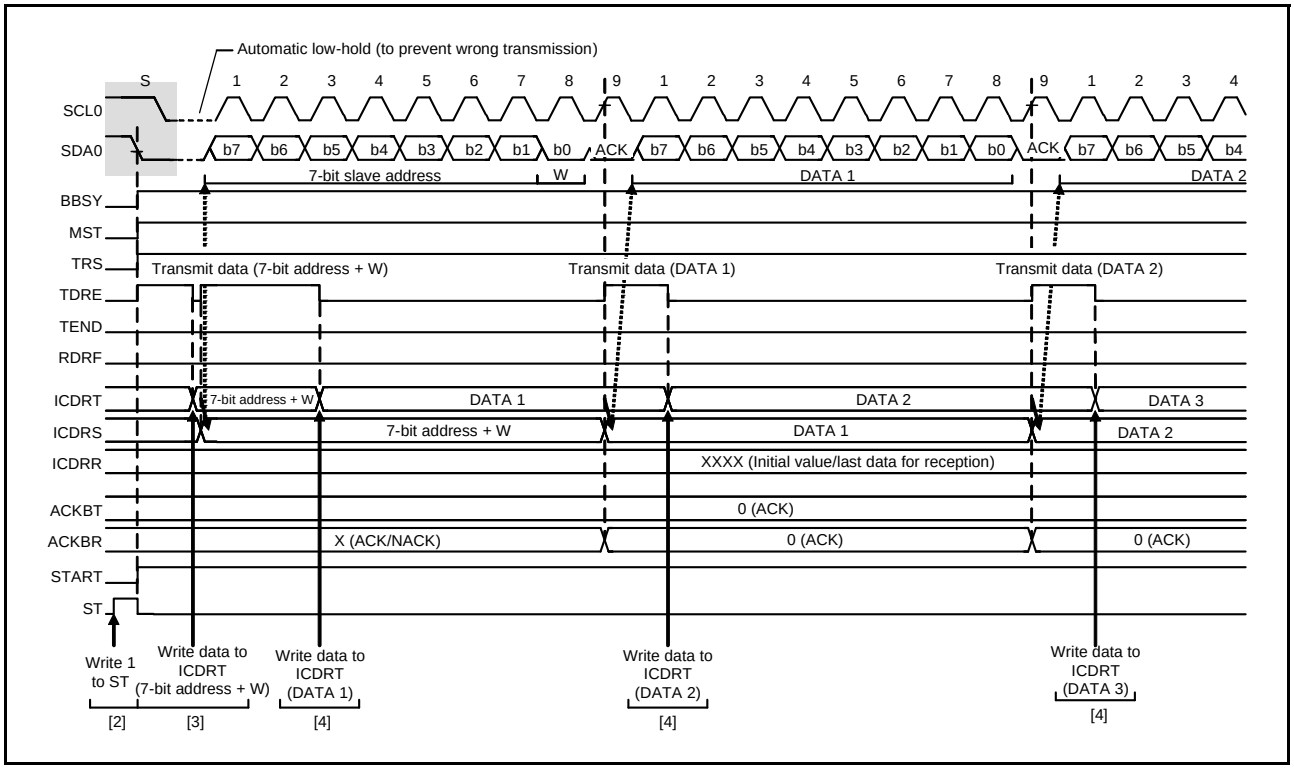



**Figure 27.7 Master Transmit Operation Timing (1) (7-Bit Address Format)**

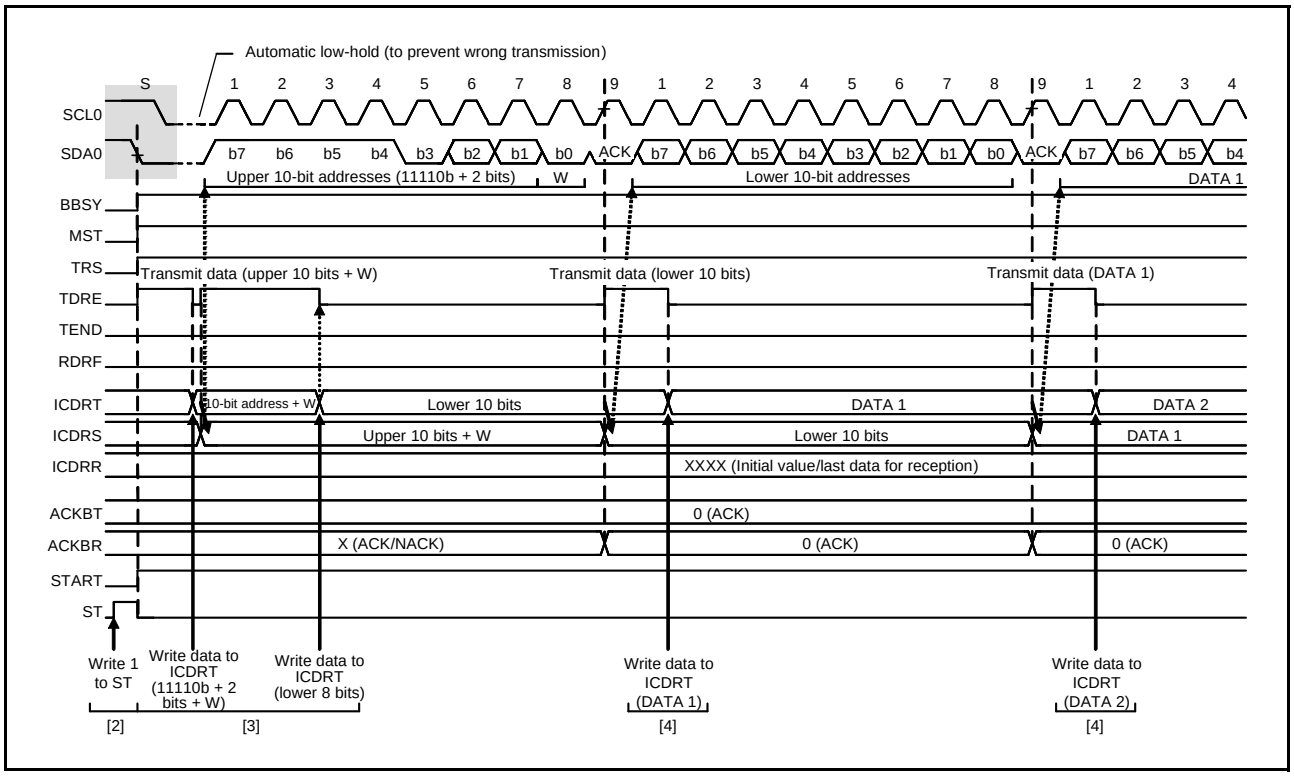



**Figure 27.8 Master Transmit Operation Timing (2) (10-Bit Address Format)**

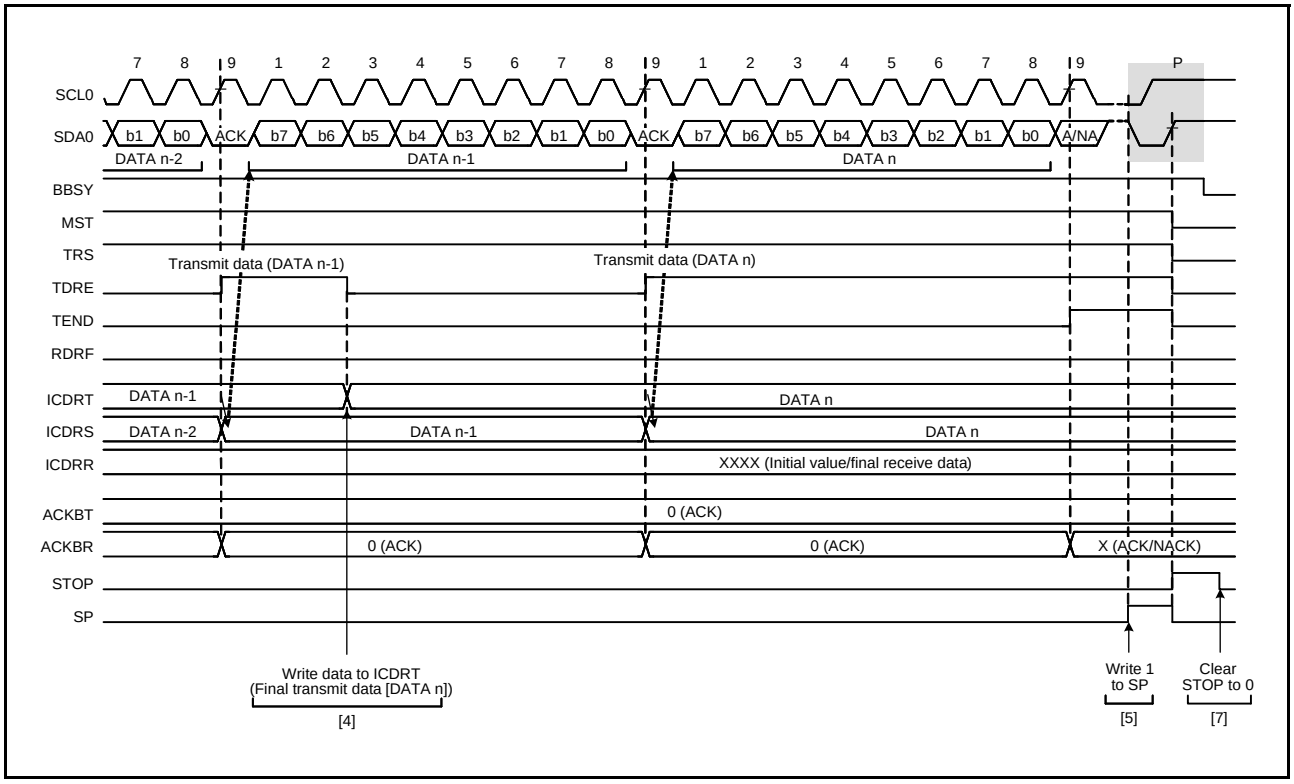



**Figure 27.9 Master Transmit Operation Timing (3)**

## 27.3.4 Master Receive Operation

In master receive operation, the RIIC as a master device outputs the SCL clock, receives data from the slave device, and returns acknowledgements. Since the RIIC must start by sending a slave address to the corresponding slave device, this part of the procedure is performed in master transmit mode, but the subsequent steps are in master receive mode.
Figure 27.11 shows an example of usage of master reception and Figure 27.12 and Figure 27.14 show the timing of operations in master reception.
The following describes the procedure and operations for master reception.

(1) Initial settings. For details, refer to section 27.3.2, Initial Settings.

(2) Read the BBSY flag in ICCR2 to check that the bus is open, and then set the ST bit in ICCR2 to 1 (start condition issuance request). Upon receiving the request, the RIIC issues a start condition. When the RIIC detects the start condition, the BBSY flag and the START flag in ICSR2 are automatically set to 1 and the ST bit is automatically cleared to 0. At this time, if the start condition is detected and the levels for the SDA output and the levels on the SDA0 line have matched while the ST bit is 1, the RIIC recognizes that issuing of the start condition as requested by the ST bit has been successfully completed, and the MST and TRS bits in ICCR2 are automatically set to 1, placing the RIIC in master transmit mode. The TDRE flag in ICSR2 is also automatically set to 1 in response to setting of the TRS bit to 1.

(3) Check that the TDRE flag in ICSR2 is 1, and then write the value for transmission (the first byte indicates the slave address and value of the R/W# bit) to ICDRT. Once the data for transmission are written to ICDRT, the TDRE flag is automatically cleared to 0, the data are transferred from ICDRT to ICDRS, and the TDRE flag is again set to 1. Once the byte containing the slave address and R/W# bit has been transmitted, the value of the ICCR2.TRS bit is automatically updated to select transmit or receive mode in accord with the value of the transmitted R/W# bit. If the value of the R/W# bit was 1, the TRS bit is cleared to 0 on the rising edge of the ninth cycle of SCL clock, placing the RIIC in master receive mode. At this time, the TDRE flag is automatically cleared to 0 and the ICSR2.RDRF

flag is automatically set to 1.
Since the ICSR2.NACKF flag being 1 at this time indicates that no slave device recognized the address or there was an error in communications, write 1 to the ICCR2.SP bit to issue a stop condition.
For master reception from a device with a 10-bit address, start by using master transmission to issue the 10-bit address, and then issue a restart condition. After that, transmitting 1111 0b, the two higher-order bits of the slave address, and the R bit places the RIIC in master receive mode.

(4) Dummy read ICDRR after confirming that the RDRF flag in ICSR2 is 1; this makes the RIIC start output of the SCL clock and start data reception.

(5) After 1 byte of data has been received, the RDRF flag in ICSR2 is set to 1 on the rising edge of the eighth or ninth cycle of SCL clock (the clock signal) as selected by the RDRFS bit in ICMR3. Reading out ICDRR at this time will produce the received data, and the RDRF flag is automatically cleared to 0 at the same time. Furthermore, the value of the acknowledgement field received during the ninth cycle of SCL clock is returned as the value set in the ACKBT bit of ICMR3. Furthermore, if the next byte to be received is the next to last byte, set the ICMR3.WAIT bit to 1 (for wait insertion) before reading the ICDRR (containing the second byte from last). As well as enabling NACK output even in the case of delays in processing to set the ICMR3.ACKBT bit to 1 (NACK) in step (6), due to other interrupts, etc., this fixes the SCL0 line to the low level on the rising edge of the ninth clock cycle in reception of the last byte, so the state is such that issuing a stop condition is possible.

(6) When the ICMR3.RDRFS bit is 0 and the slave device must be notified that it is to end transfer for data reception after transfer of the next (final) byte, set the ACKBT bit in ICMR3 to 1 (NACK).

(7) After reading out the byte before last from the ICDRR register, if the value of the ICSR2.RDRF flag is confirmed to be 1, write 1 to the SP bit in ICCR2 (stop condition issuance request) and then read the last byte from ICDRR. When ICDRR is read, the RIIC is released from the wait state and issues the stop condition after low-level output in the ninth clock cycle is completed or the SCL0 line is released from the low-hold state.

(8) Upon detecting the stop condition, the RIIC automatically clears the MST and TRS bits in ICCR2 to 00b and enters slave receive mode. Furthermore, detection of the stop condition leads to setting of the STOP flag in ICSR2 to 1.

(9) After checking that the ICSR2.STOP flag is 1, clear the ICSR2.NACKF and STOP flags to 0 for the next transfer operation.

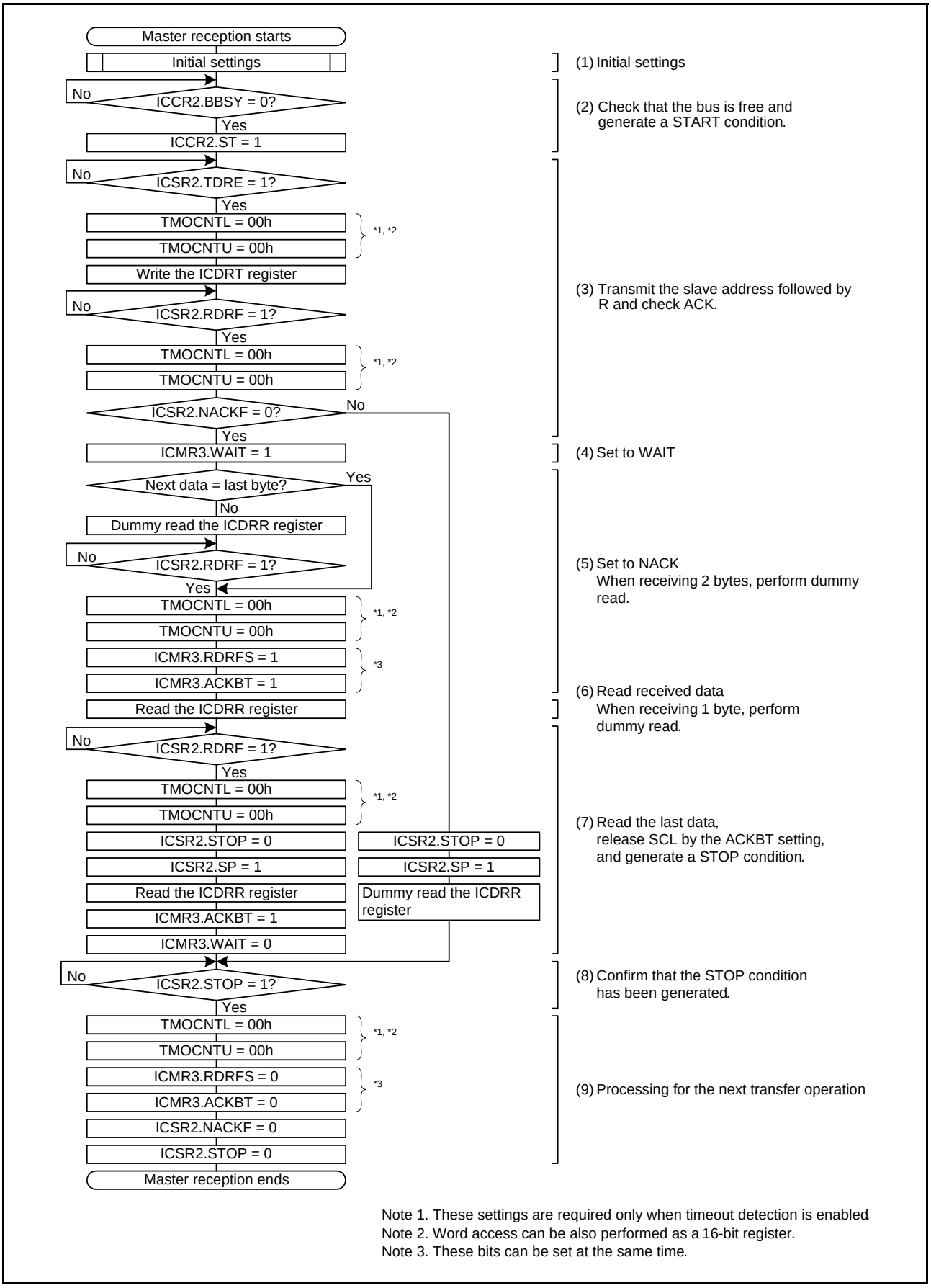
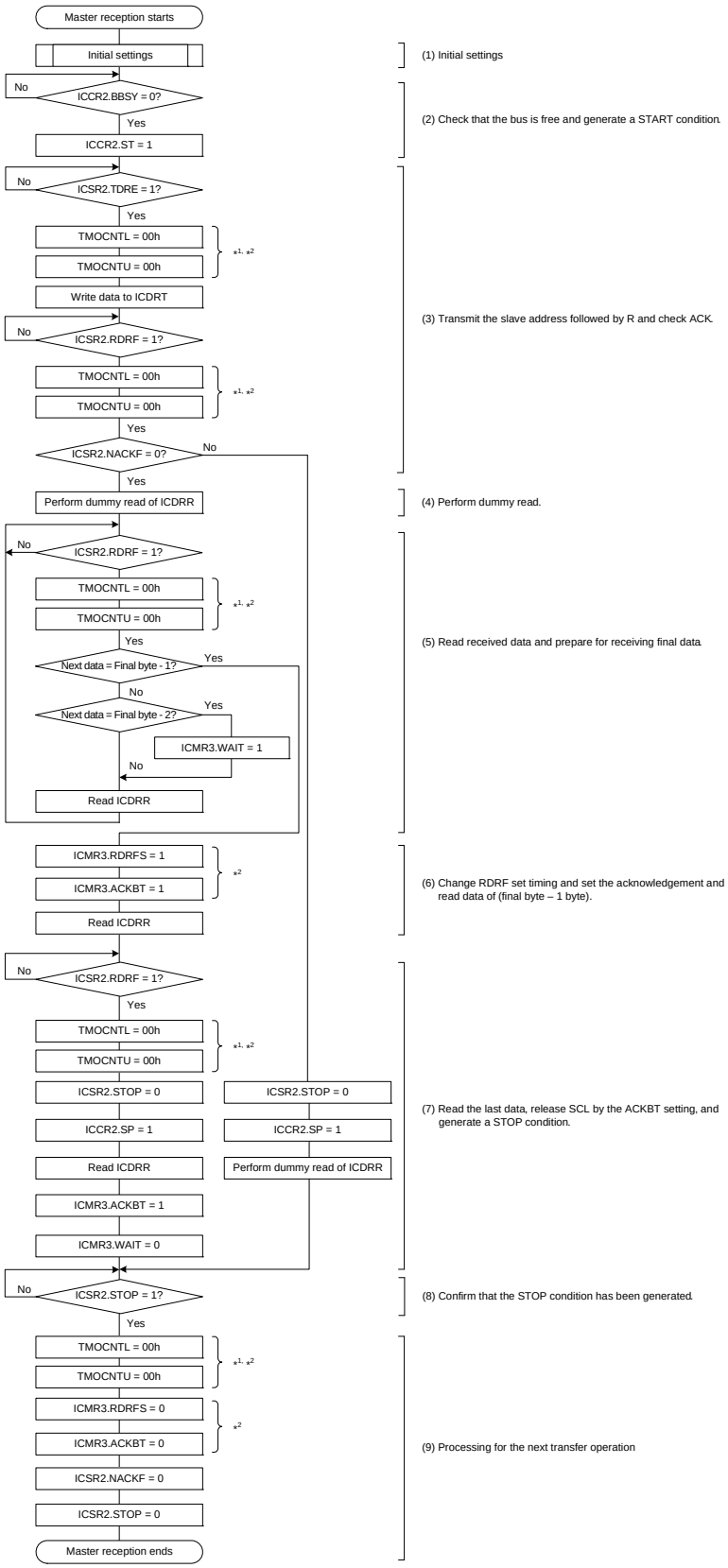



**Figure 27.10 Example of Master Reception (7-Bit Address Format, 1 or 2 bytes)**




Note 1. These settings are required only when timeout detection is enabled.
Note 2. Word access can be also performed as a 16-bit register.
Note 3. These bits can be set at the same time.

**Figure 27.11 Example of Master Reception (7-Bit Address Format, 3 Bytes or More)**

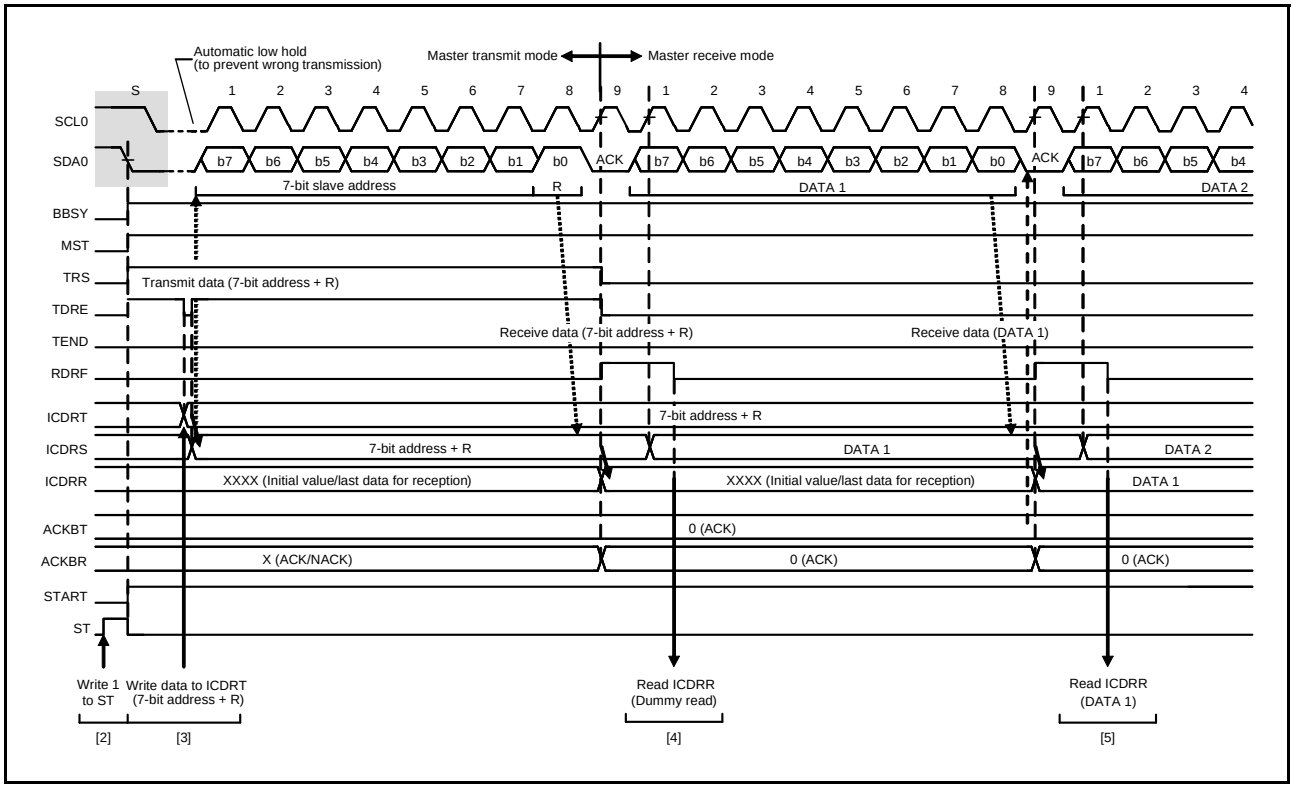



**Figure 27.12 Master Receive Operation Timing (1) (7-Bit Address Format, when RDRFS = 0)**

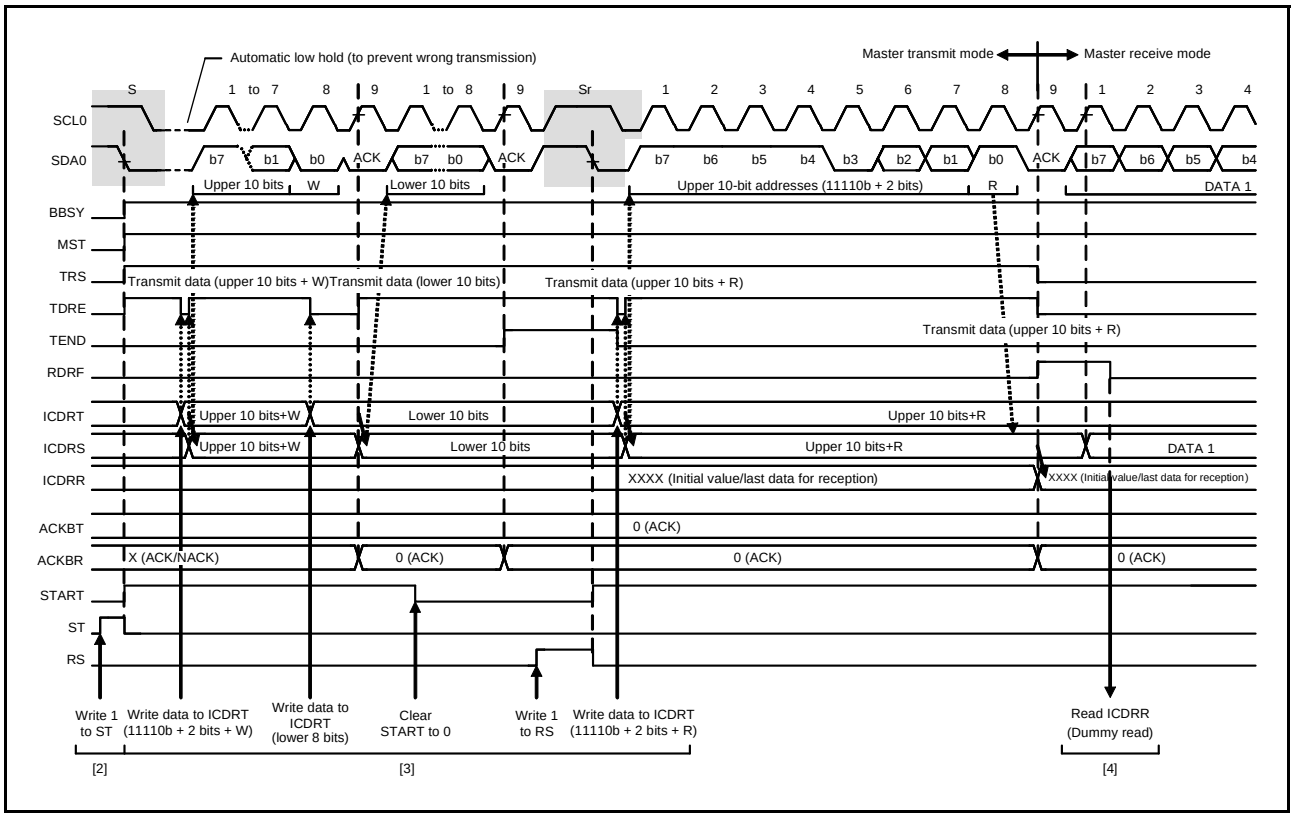



**Figure 27.13 Master Receive Operation Timing (2) (10-Bit Address Format, when RDRFS = 0)**

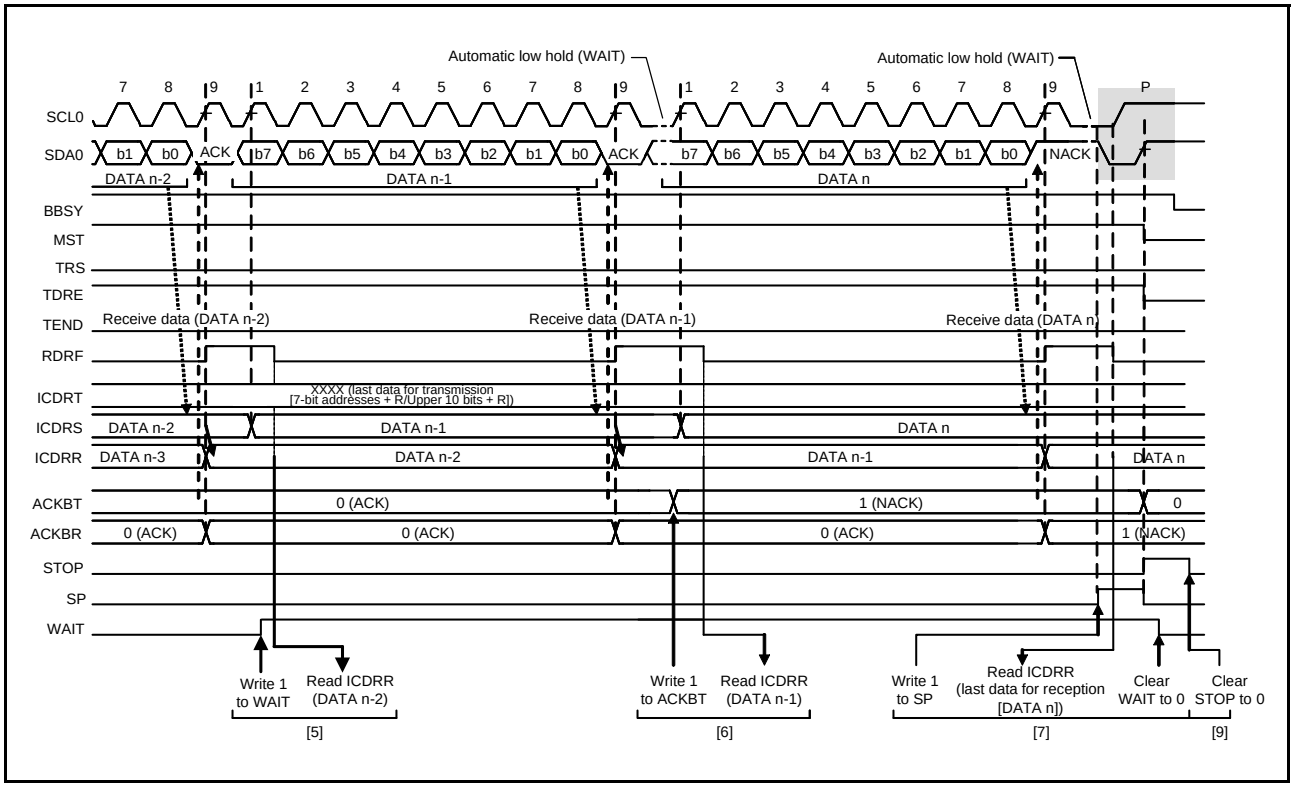
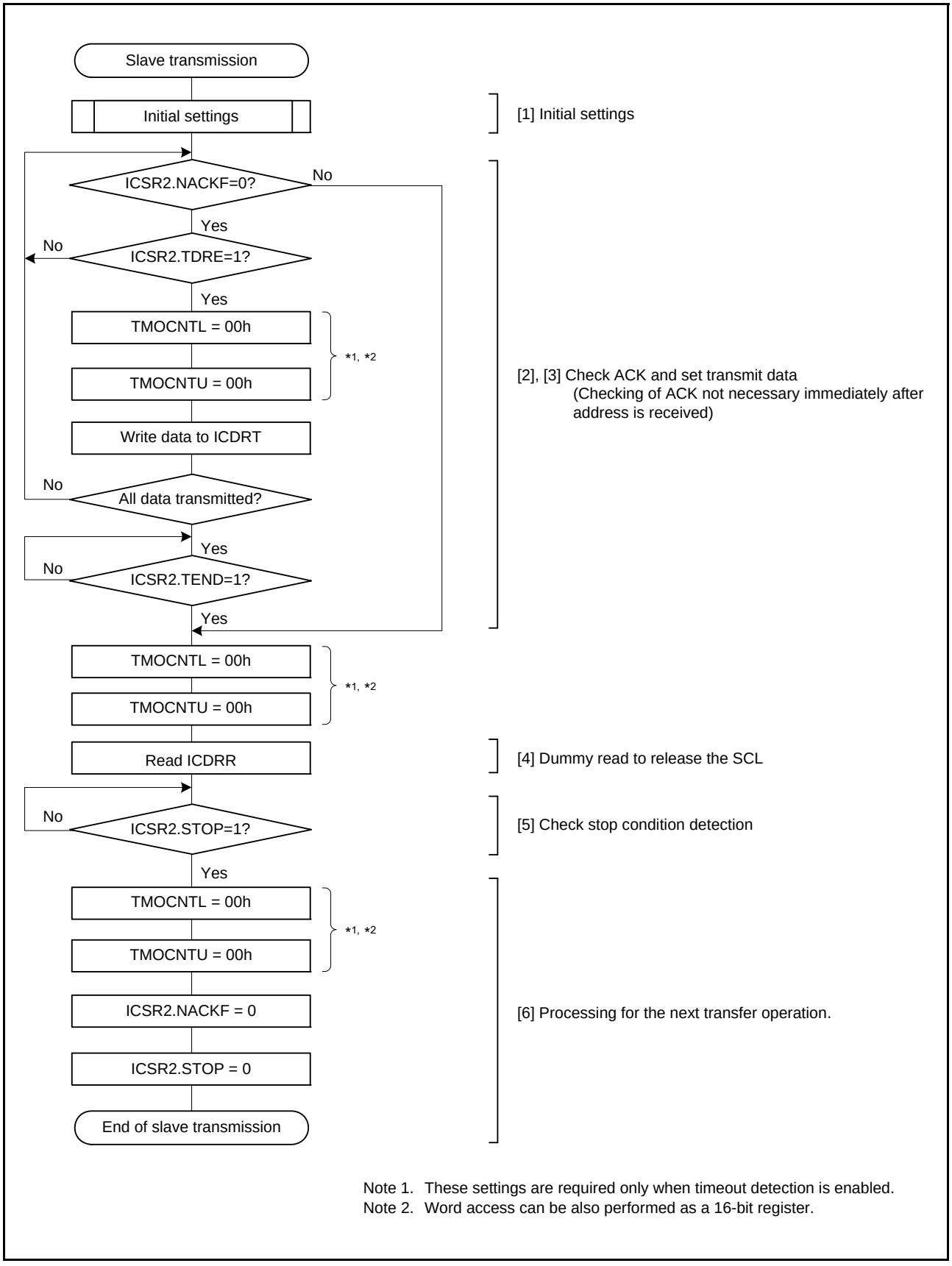



**Figure 27.14 Master Receive Operation Timing (3) (when RDRFS = 0)**

### 27.3.5 Slave Transmit Operation

In slave transmit operation, the master device outputs the SCL clock, the RIIC transmits data as a slave device, and the master device returns acknowledgements.
Figure 27.15 shows an example of usage of slave transmission and Figure 27.16 and Figure 27.17 show the timing of operations in slave transmission.
The following describes the procedure and operations for slave transmission.

(1) Initial settings. For details, refer to section 27.3.2, Initial Settings.
After initial settings, the RIIC will stay in the standby state until it receives a slave address that it matches.
(2) After receiving a matching slave address, the RIIC sets one of the corresponding bits ICSR1.HOA, GCA, and AASy (y = 0 to 2) to 1 on the rising edge of the ninth cycle of SCL clock (the clock signal) and outputs the value set in the ICMR3.ACKBT bit to the acknowledge bit on the ninth cycle of SCL clock. If the value of the R/W# bit that was also received at this time is 1, the RIIC automatically places itself in slave transmit mode by setting both the TRS bit and the TDRE flag in ICSR2 to 1.
(3) After the ICSR2.TEND flag is confirmed to be 1, write the data for transmission to the ICDRT register. At this time, if the RIIC receives no acknowledge from the master device (receives an NACK signal) while the ICFER.NACKE bit is 1, the RIIC suspends transfer of the next data.
(4) Wait unit the ICSR2.TEND flag is set to 1 while the ICSR2.TDRE flag is 1, after the ICSR2.NACKF flag is set to 1 or the last byte for transmission is written to the ICDRT register. When the ICSR2.NACKF flag or the TEND flag is 1, the RIIC drives the SCL0 line low on the ninth falling edge of SCL clock.
(5) When the ICSR2.NACKF flag or the ICSR2.TEND flag is 1, dummy read ICDRR to complete the processing. This releases the SCL0 line.
(6) Upon detecting the stop condition, the RIIC automatically clears bits ICSR1.HOA, GCA, and AASy (y = 0 to 2), flags ICSR2.TDRE and TEND, and the ICCR2.TRS bit to 0, and enters slave receive mode.
(7) After checking that the ICSR2.STOP flag is 1, clear the ICSR2.NACKF and STOP flags to 0 for the next transfer operation.




Note 1. These settings are required only when timeout detection is enabled.
Note 2. Word access can be also performed as a 16-bit register.

**Figure 27.15 Example of Slave Transmission Flowchart**

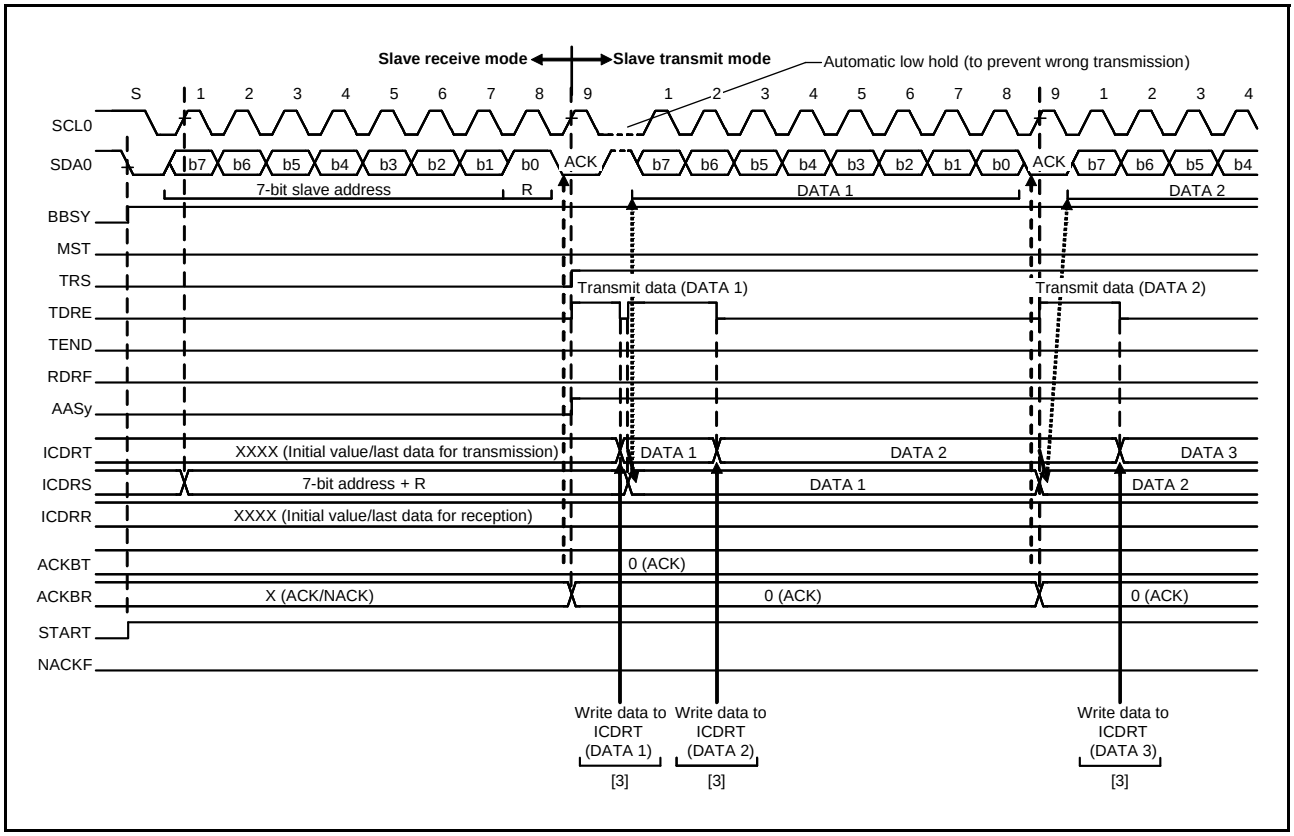



**Figure 27.16 Slave Transmit Operation Timing (1) (7-Bit Address Format)**

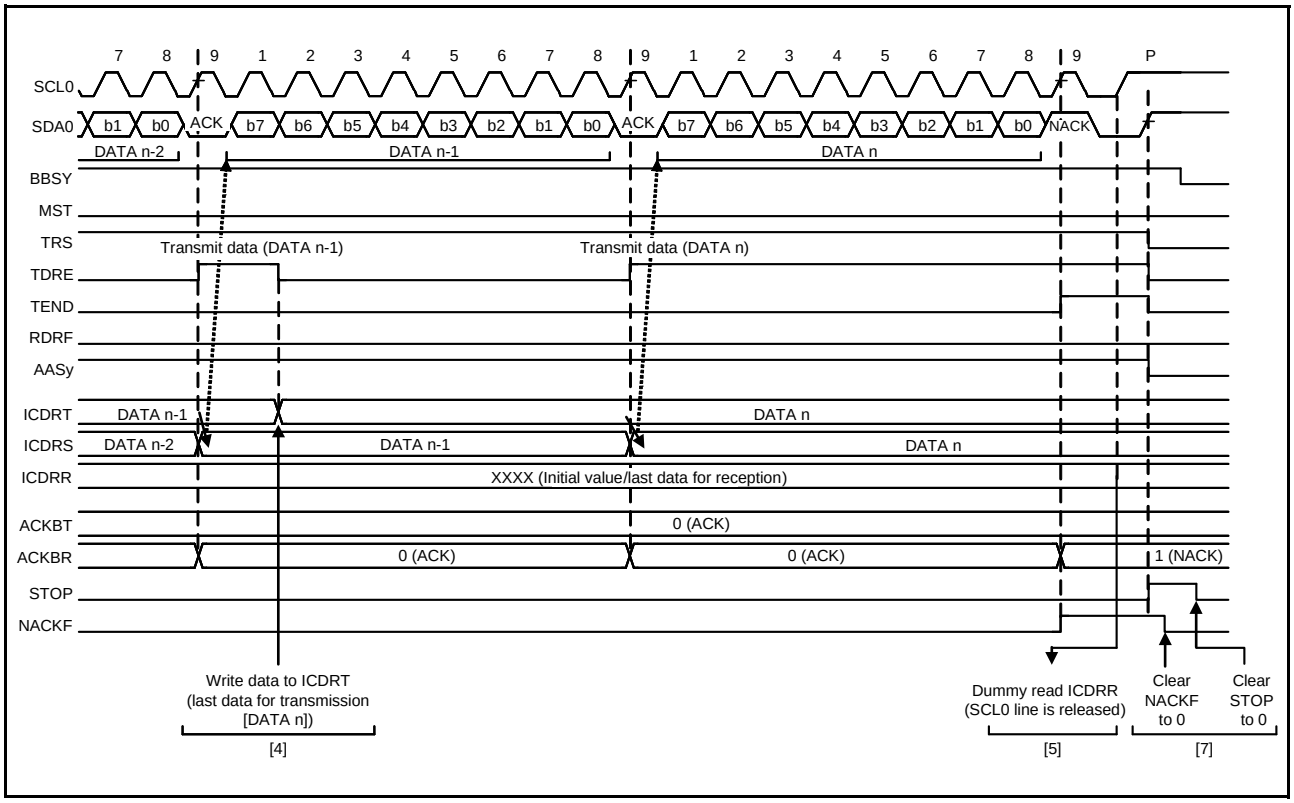



**Figure 27.17 Slave Transmit Operation Timing (2)**

### 27.3.6 Slave Receive Operation

In slave receive operation, the master device outputs the SCL clock and transmit data, and the RIIC returns acknowledgements as a slave device.
Figure 27.18 shows an example of usage of slave reception and Figure 27.19 and Figure 27.20 show the timing of operations in slave reception.
The following describes the procedure and operations for slave reception.

(1) Initial settings. For details, refer to section 27.3.2, Initial Settings.
After initial settings, the RIIC will stay in the standby state until it receives a slave address that it matches.
(2) After receiving a matching slave address, the RIIC sets one of the corresponding bits ICSR1.HOA, GCA, and AASy (y = 0 to 2) to 1 on the rising edge of the ninth cycle of SCL clock (the clock signal) and outputs the value set in the ICMR3.ACKBT bit to the acknowledge bit on the ninth cycle of SCL clock. If the value of the R/W# bit that was also received at this time is 0, the RIIC continues to place itself in slave receive mode and sets the RDRF flag in ICSR2 to 1.
(3) After the ICSR2.STOP flag is confirmed to be 0 and the ICSR2.RDRF flag to be 1, dummy read ICDRR (the dummy value consists of the slave address and R/W# bit when the 7-bit address format is selected, or the lower 8 bits when the 10-bit address format is selected).
(4) When ICDRR is read, the RIIC automatically clears the ICSR2.RDRF flag to 0. If reading of ICDRR is delayed and a next byte is received while the RDRF flag is still set to 1, the RIIC holds the SCL0 line low from one SCL cycle before the timing with which RDRF should be set. In this case, reading ICDRR releases the SCL0 line from being held at the low level.
When the ICSR2.STOP flag is 1 and the ICSR2.RDRF flag is also 1, read ICDRR until all the data is completely received.
(5) Upon detecting the stop condition, the RIIC automatically clears bits ICSR1.HOA, GCA, and AASy (y = 0 to 2) to 0.
(6) After checking that the ICSR2.STOP flag is 1, clear the ICSR2.STOP flag to 0 for the next transfer operation.

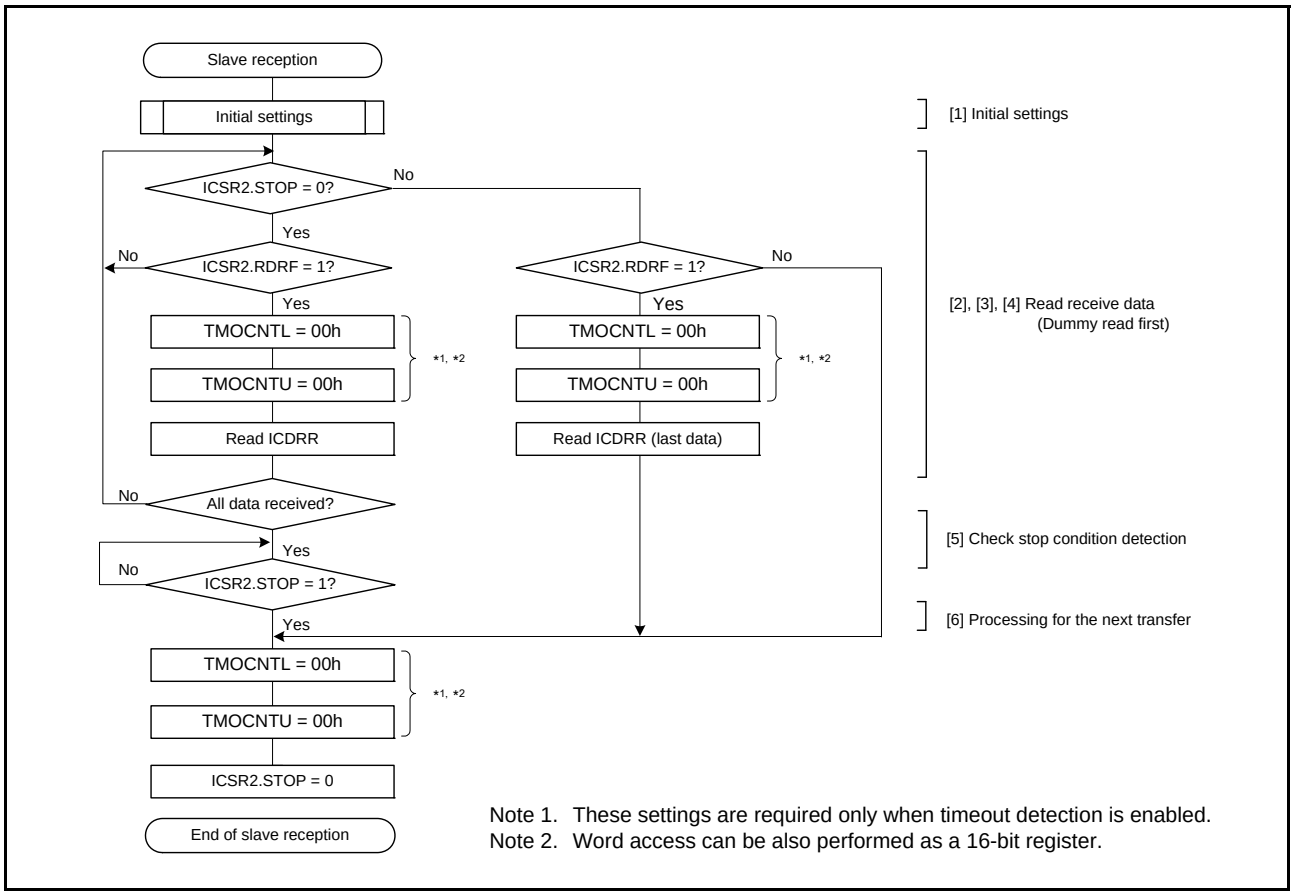



**Figure 27.18 Example of Slave Reception Flowchart**

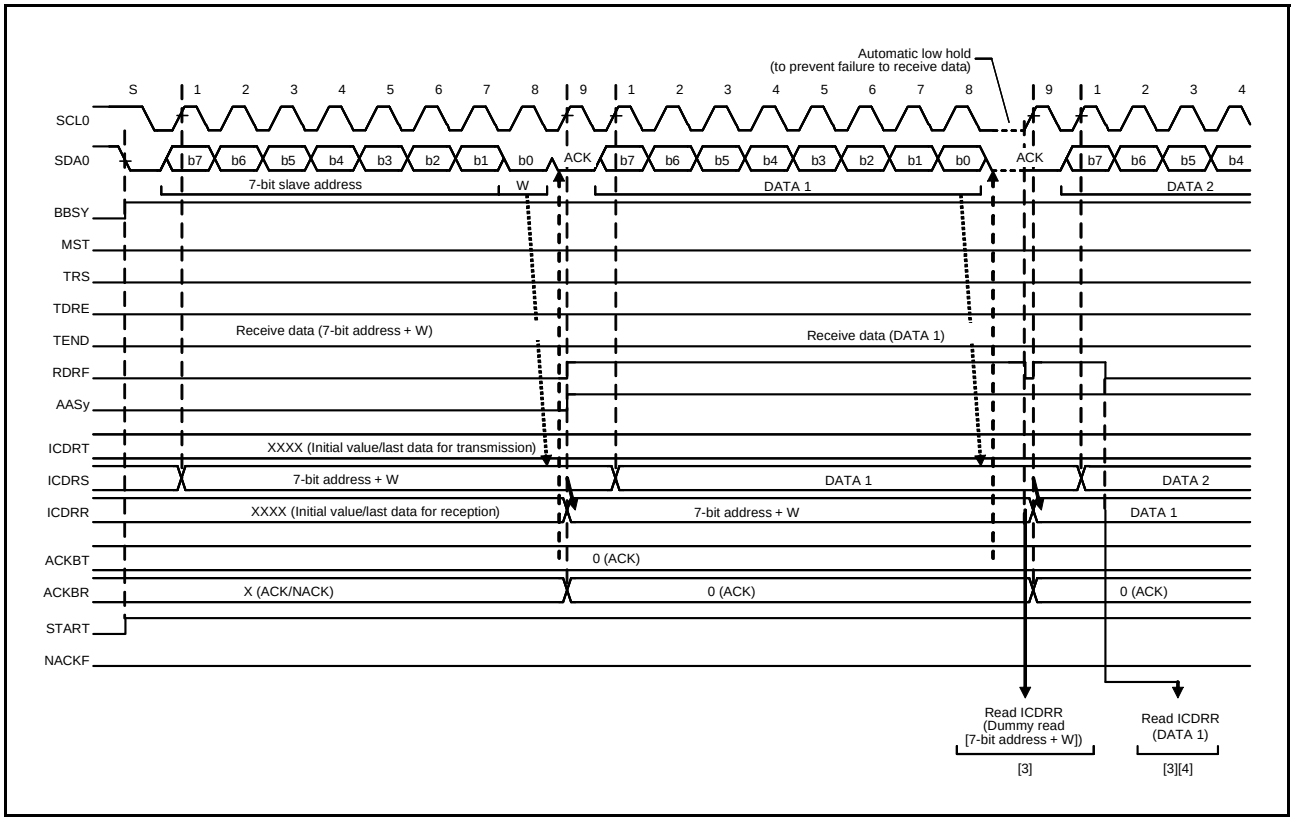



**Figure 27.19 Slave Receive Operation Timing (1) (7-Bit Address Format, when RDRFS = 0)**

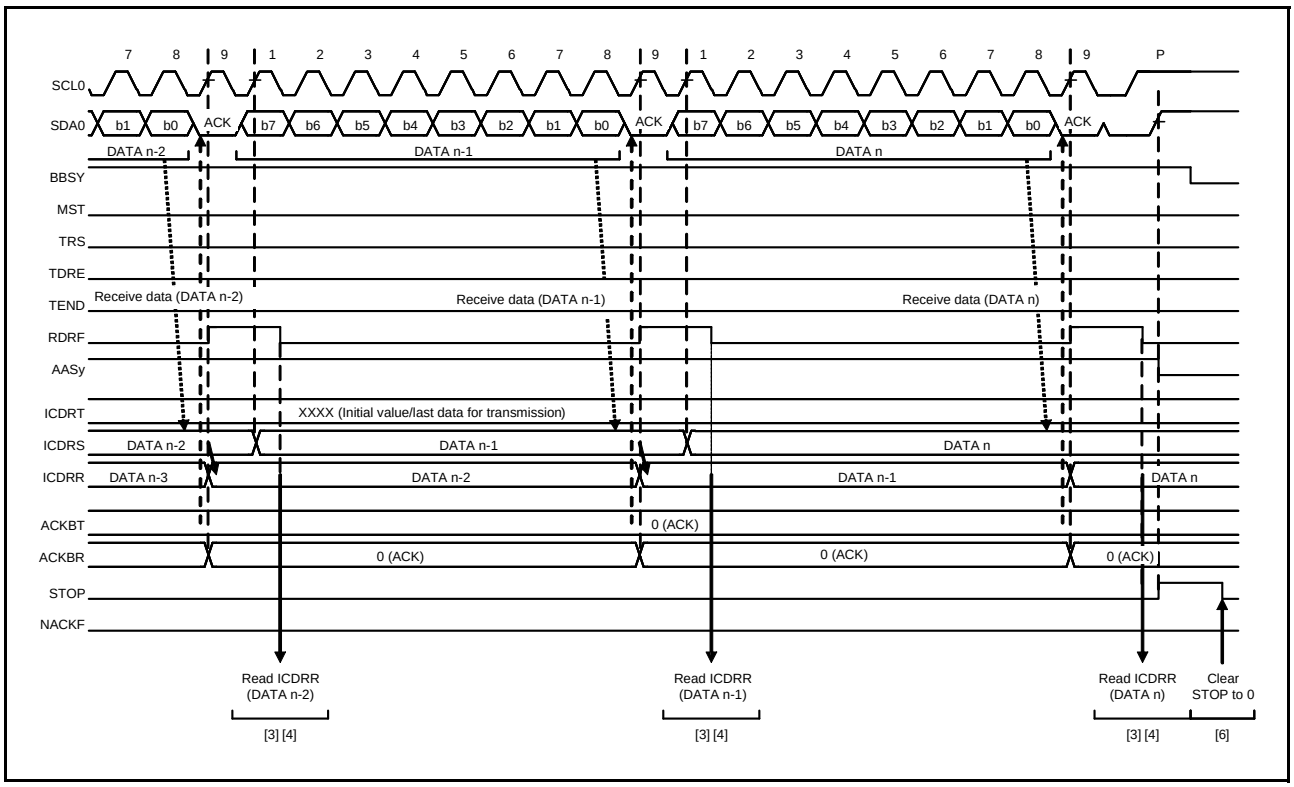



**Figure 27.20 Slave Receive Operation Timing (2) (when RDRFS = 0)**

## 27.4 SCL Synchronization Circuit

In generation of the SCL clock, the RIIC starts counting out the value for width at high level specified in ICBRH when it detects a rising edge on the SCL0 line and drives the SCL0 line low once counting of the width at high level is complete. When the RIIC detects the falling edge of the SCL0 line, it starts counting out the width at low level period specified in ICBRL, and then stops driving the SCL0 line (releases the line) once counting of the width at low level is complete. The SCL clock is thus generated.

If multiple master devices are connected to the I2C bus, a collision of SCL signals may arise due to contention with another master device. In such cases, the master devices have to synchronize their SCL signals. Since this synchronization of SCL signals must be bit by bit, the RIIC is equipped with a facility (the SCL synchronization circuit) to obtain bit-by-bit synchronization of the SCL clock signals by monitoring the SCL0 line while in master mode.

When the RIIC has detected a rising edge on the SCL0 line and thus started counting out the width at high level specified in ICBRH, and the level on the SCL0 line falls because an SCL signal is being generated by another master device, the RIIC stops counting when it detects the falling edge, drives the level on the SCL0 line low, and starts counting out the width at low level specified in ICBRL. When the RIIC finishes counting out the width at low level, it stops driving the SCL0 line to the low level (i.e. releases the line). At this time, if the width at low level of the SCL clock signal from the other master device is longer than the width at low level set in the RIIC, the width at low level of the SCL signal will be extended. Once the width at low level for the other master device has ended, the SCL signal rises because the SCL0 line has been released. When the RIIC finishes outputting the low-level period of the SCL clock, the SCL0 line is released and the SCL clock rises. That is, in cases of contention of SCL signals from more than one master, the width at high level of the SCL signal is synchronized with that of the clock having the narrower width, and the width at low level of the SCL signal is synchronized with that of the clock having the broader width. However, such synchronization of the SCL signal is only enabled when the SCLE bit in ICFER is set to 1.

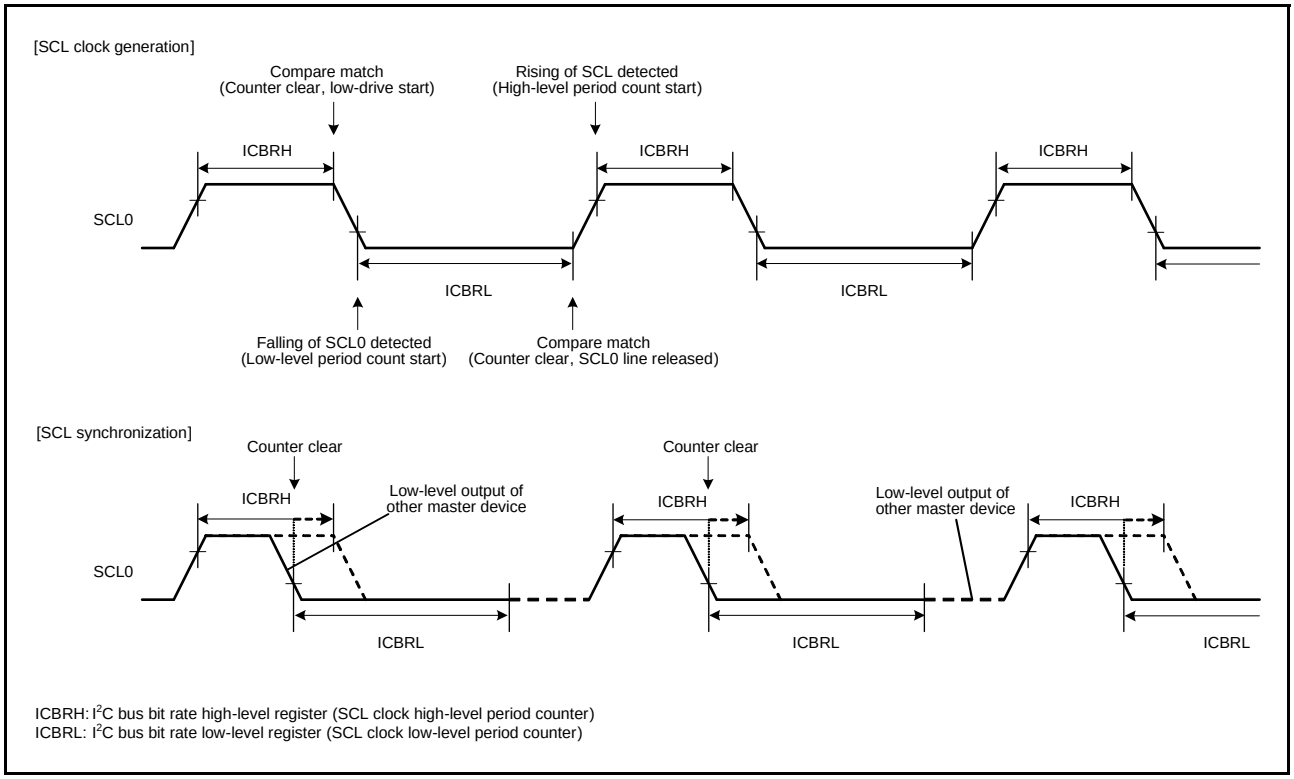



**Figure 27.21 Generation and Synchronization of the SCL Signal from the RIIC**

## 27.5 Facility for Delaying SDA Output

The RIIC module incorporates a facility for delaying output on the SDA line. The delay can be applied to all output (issuing of the start, restart, and stop conditions, data, and the ACK and NACK signals) on the SDA line.

With the SDA output delay facility, SDA output is delayed from detection of a falling edge of the SCL signal to ensure that the SDA signal is output within the interval over which the SCL clock is at the low level. Doing this leads to usage with the aim of preventing erroneous operation of communications devices, with the aim of satisfying the 300-ns (min.) data-hold time requirement of the SMBus specification.

The output delay facility is enabled by setting the SDDL[2:0] bits in ICMR2 to any value other than 000b, and disabled by setting the same bits to 000b.

While the SDA output delay facility is enabled (i.e. while the SDDL[2:0] bits in ICMR2 are set to any value other than 000b), the DLCS bit in ICMR2 selects the clock source for counting by the SDA output delay counter as the internal base clock (IICϕ) for the RIIC module or as a clock signal derived by dividing the frequency of the internal base clock by two (IICϕ/2). The counter counts the number of cycles set in the SDDL[2:0] bits in ICMR2. After counting of the set number of cycles of delay is completed, the RIIC module places the required output (start, restart, or stop condition, data, or an ACK or NACK signal) on the SDA line.

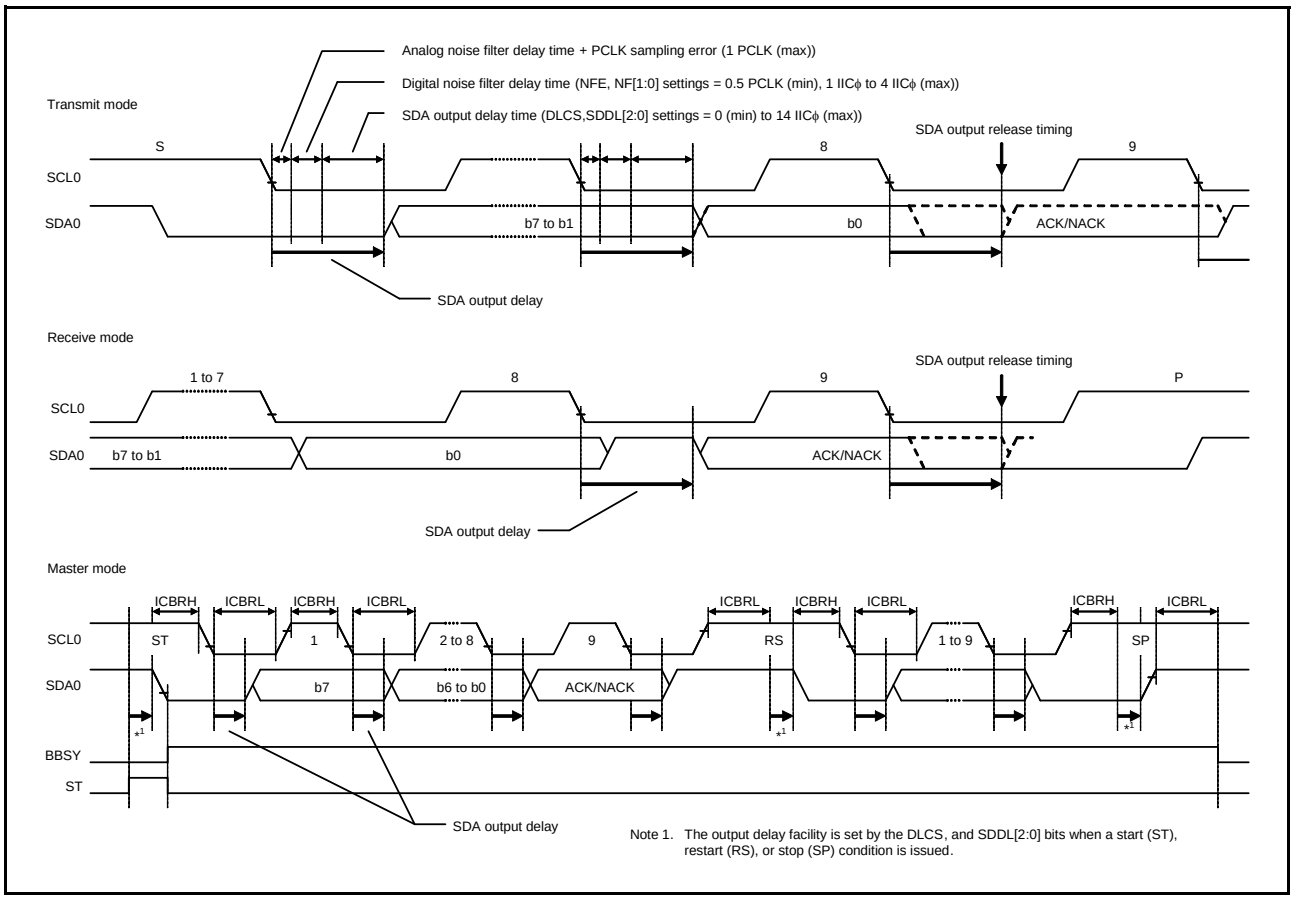



**Figure 27.22 SDA Output Delay Facility**

## 27.6 Digital Noise-Filter Circuits

The states of the SCL0 and SDA0 pins are conveyed to the internal circuitry through analog noise-filter and digital noise-filter circuits. Figure 27.23 is a block diagram of the digital noise-filter circuit.

The on-chip digital noise-filter circuit of the RIIC consists of four flip-flop circuit stages connected in series and a match-detection circuit.

The number of effective stages in the digital noise filter is selected by the NF[1:0] bits in ICMR3. The selected number of effective stages determines the noise-filtering capability as a period from one to four IICϕ cycles.

The input signal to the SCL0 pin (or SDA0 pin) is sampled on falling edges of the IICϕ signal. When the input signal level matches the output level of the number of effective flip-flop circuit stages as selected by the NF[1:0] bits in ICMR3, the signal level is conveyed to the subsequent stage. If the signal levels do not match, the previous value is retained.

If the ratio between the frequency of the internal operating clock (PCLK) and the transfer rate is small (e.g. data transfer at 400 kbps with PCLK = 4 MHz), the characteristics of the digital noise filter may lead to the elimination of needed signals as noise. In such cases, it is possible to disable the digital noise-filter circuit (by clearing the NFE bit in ICFER) and use only the analog noise-filter circuit.

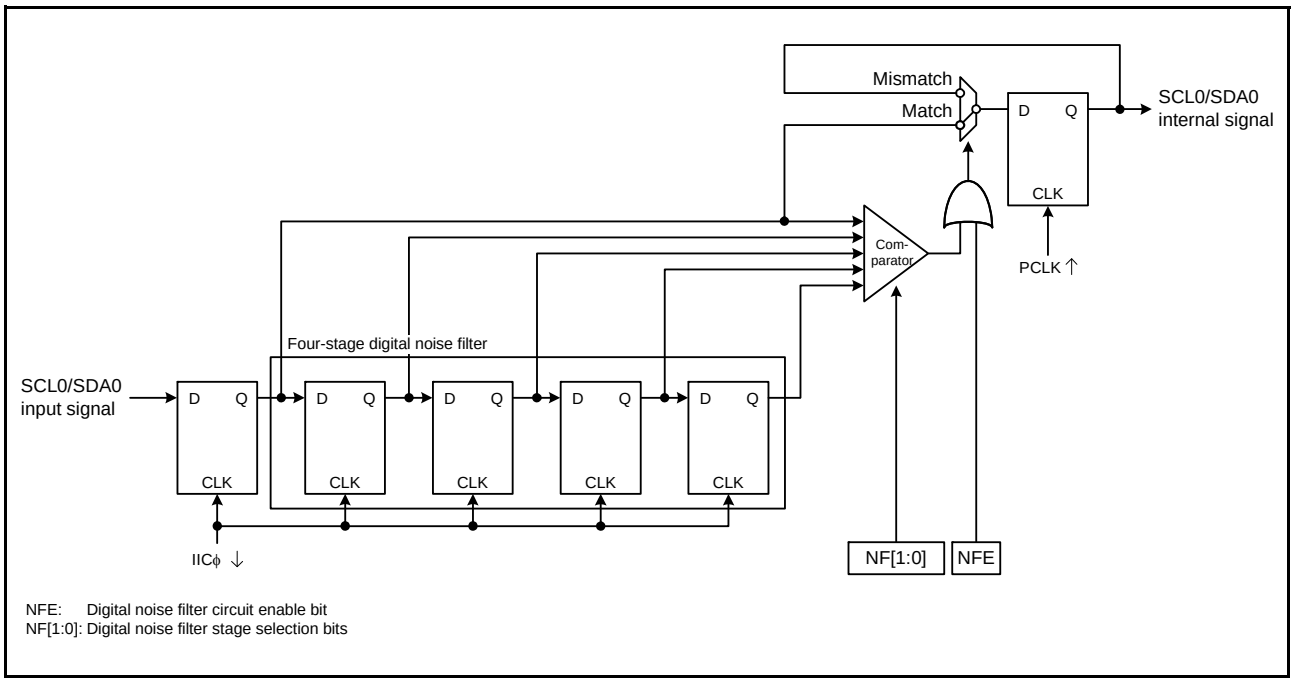



**Figure 27.23 Block Diagram of Digital Noise Filter Circuit**

## 27.7 Address Match Detection

The RIIC can set three unique slave addresses in addition to the general call address and host address, and also can set 7-bit or 10-bit slave addresses.

### 27.7.1 Slave-Address Match Detection

The RIIC can set three unique slave addresses, and has a slave address detection function for each unique slave address. When the SARyE bit (y = 0 to 2) in ICSER is set to 1, the slave addresses set in SARUy and SARLy (y = 0 to 2) can be detected.

When the RIIC detects a match of the set slave address, the corresponding AASy flag (y = 0 to 2) in ICSR1 is set to 1 at the rising edge of the ninth SCL clock cycle, and the RDRF flag in ICSR2 or the TDRE flag in ICSR2 is set to 1 by the following R/W# bit. This causes a receive data full interrupt (RXI) or transmit data empty interrupt (TXI) to be generated. The AASy flag is used to identify which slave address has been specified.

Figure 27.24 to Figure 27.26 show the AASy flag set timing in three cases.

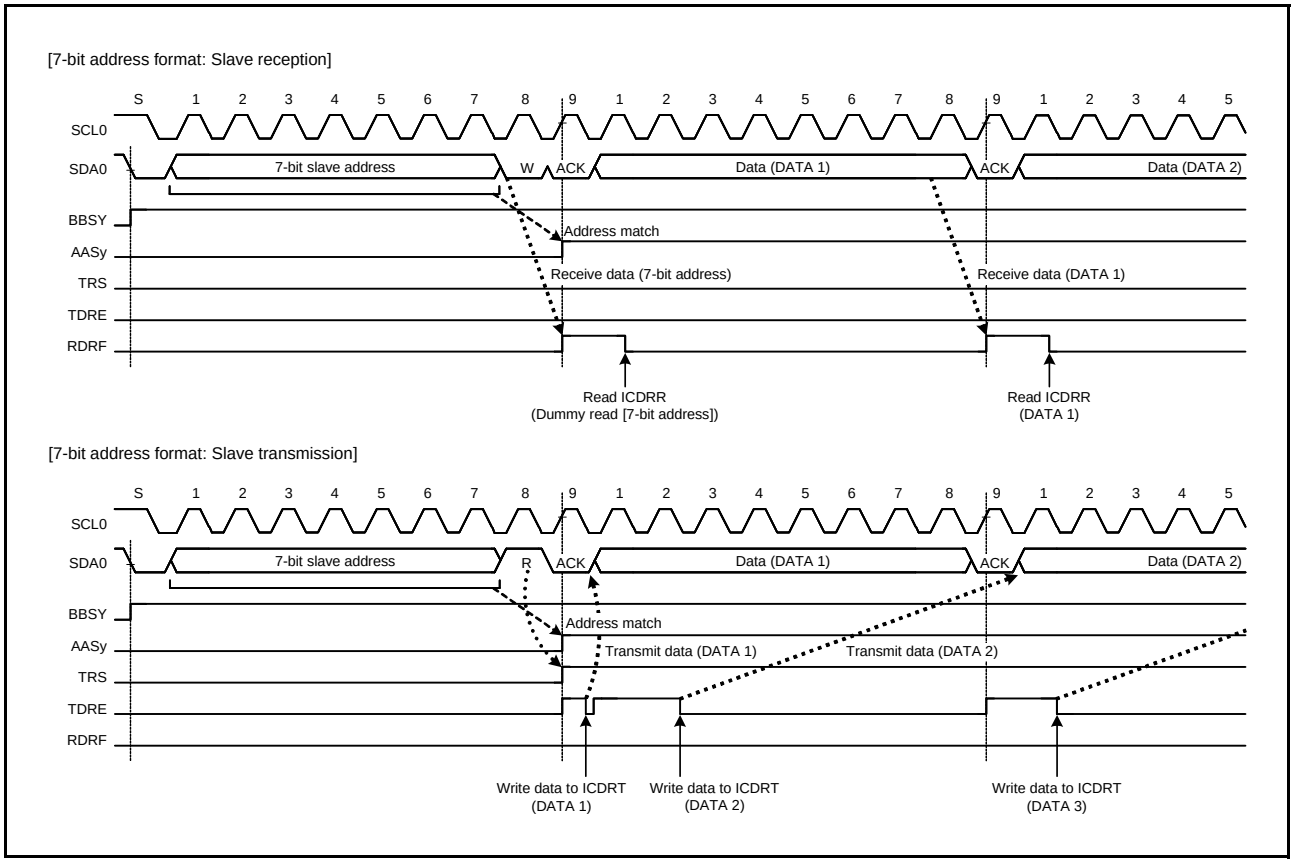



**Figure 27.24 AASy Flag Set Timing with 7-Bit Address Format Selected**

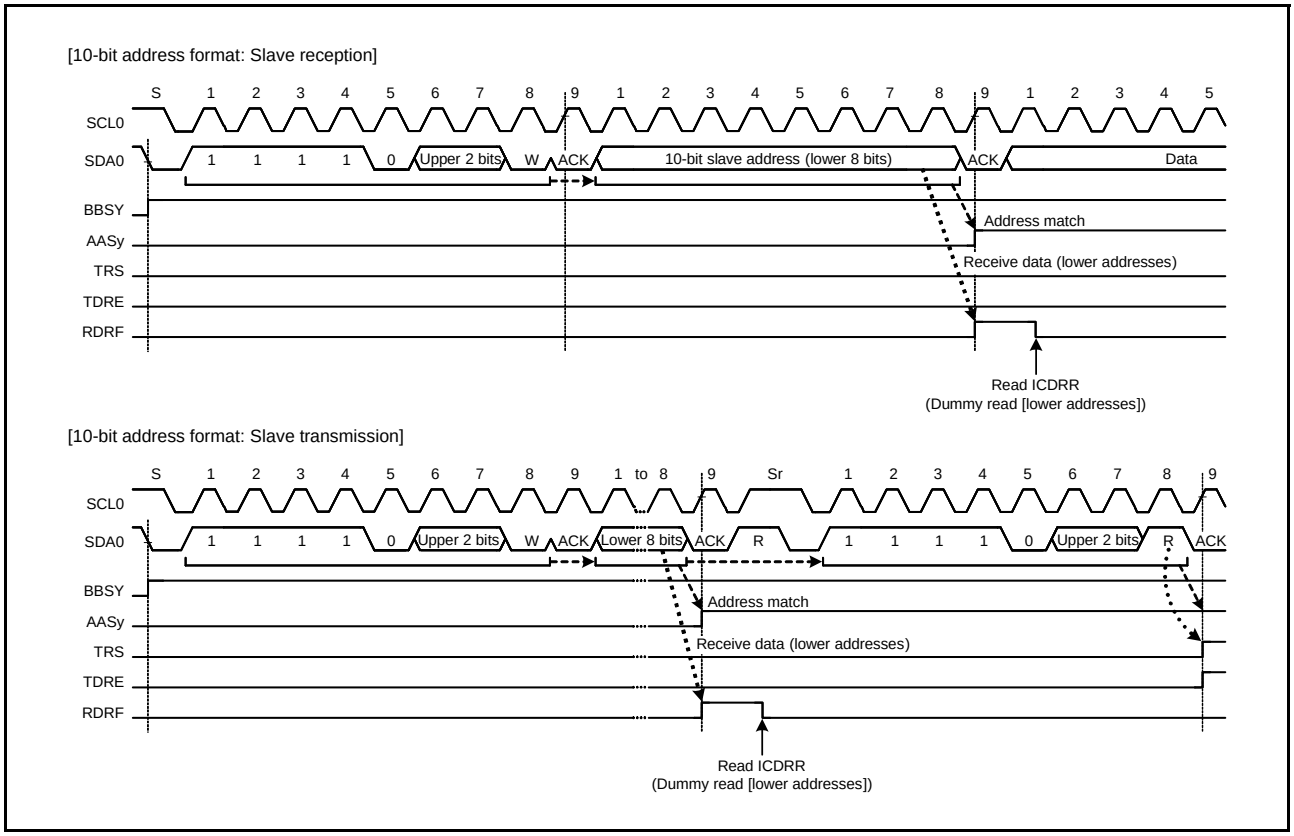



**Figure 27.25 AASy Flag Set Timing with 10-Bit Address Format Selected**

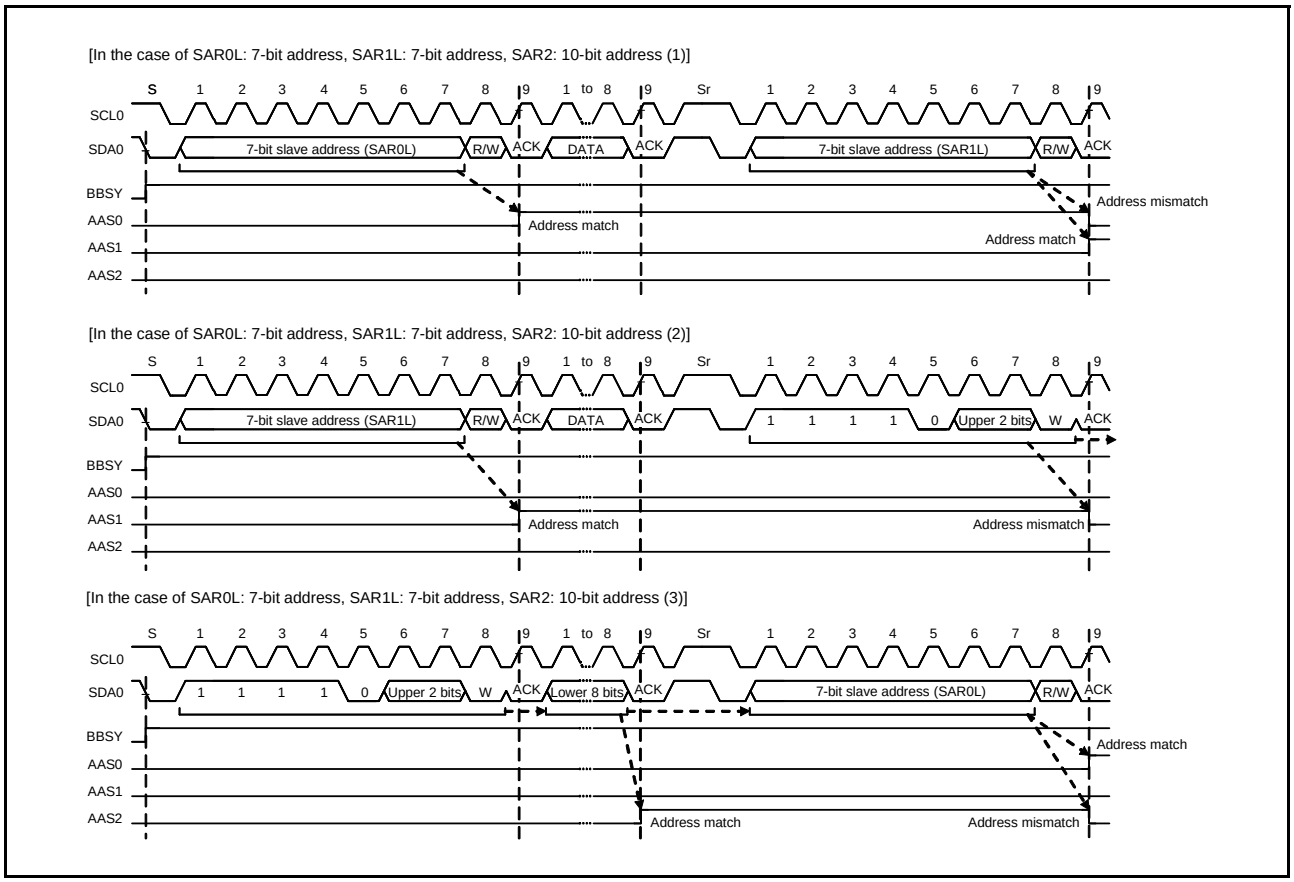



**Figure 27.26 AASy Flag Set/Clear Timing with 7-Bit/10-Bit Address Formats Mixed**

## 27.7.2 Detection of the General Call Address

The RIIC has a facility for detecting the general call address (0000 000b + 0 [W]). This is enabled by setting the GCAE bit in ICSER to 1.

If the address received after a start or restart condition is issued is 0000 000b + 1[R] (start byte), the RIIC recognizes this as the address of a slave device with an “all-zero” address but not as the general call address.

When the RIIC detects the general call address, both the GCA flag in ICSR1 and the RDRF flag in ICSR2 are set to 1 on the rising edge of the ninth cycle of SCL clock. This leads to the generation of a receive data full interrupt (RXI). The value of the GCA flag can be confirmed to recognize that the general call address has been transmitted.

Operation after detection of the general call address is the same as normal slave receive operation.

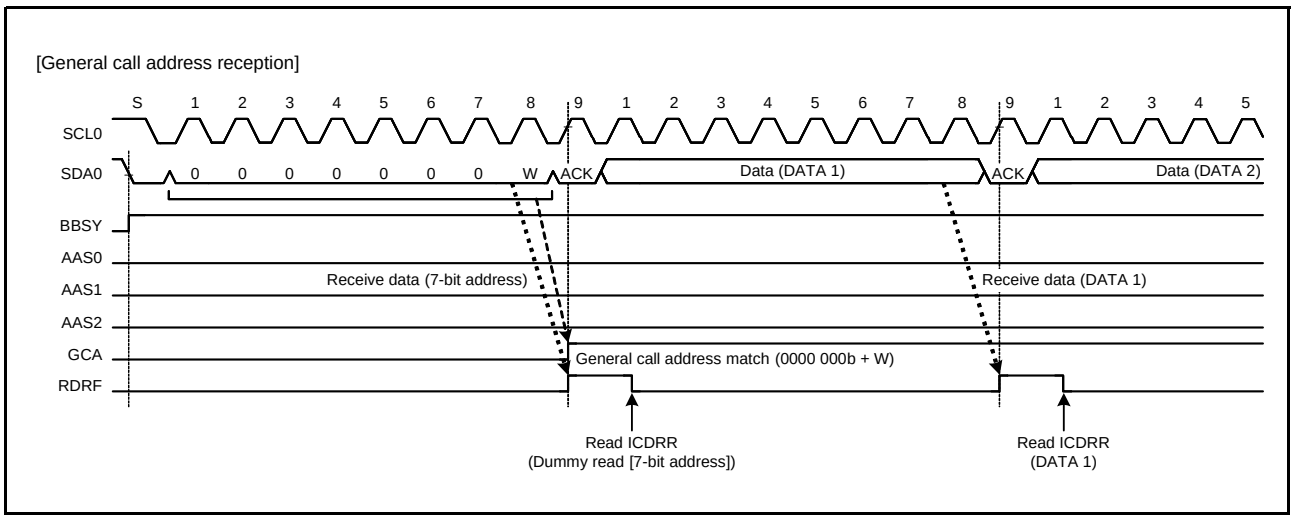



**Figure 27.27 Timing of GCA Flag Setting during Reception of General Call Address**

### 27.7.3 Device-ID Address Detection

The RIIC module has a facility for detecting device-ID addresses conformant with the I2C bus specification (Rev. 03). When the RIIC receives 1111 100b as the first byte after a start condition or restart condition was issued with the DIDE bit in ICSER set to 1, the RIIC recognizes the address as a device ID, sets the DID flag in ICSR1 to 1 on the rising edge of the eighth SCL clock cycle when the following R/W# bit is 0, and then compares the second and subsequent bytes with its own slave address. If the address matches the value in the slave address register, the RIIC sets the corresponding AASy flag (y = 0 to 2) in ICSR1 to 1.

After that, when the first byte received after a start or restart condition is issued matches the device ID address (1111 100b) again and the following R/W# bit is 1, the RIIC does not compare the second and subsequent bytes and sets the ICSR2.TDRE flag to 1.

In the device-ID address detection function, the RIIC clears the DID flag to 0 if a match with the RIIC's own slave address is not obtained or a match with the device ID address is not obtained after a match with the RIIC's own slave address and the detection of a restart condition. If the first byte after detection of a start or restart condition matches the device ID address (1111 100b) and the R/W# bit is 0, the RIIC sets the DID flag to 1 and compares the second and subsequent bytes with the RIIC's slave address. If the R/W# bit is 1, the DID flag holds the previous value and the RIIC does not compare the second and subsequent bytes. Therefore, the reception of a device-ID address can be checked by reading the DID flag after confirming that TDRE = 1.

Furthermore, prepare the device-ID fields (three bytes: 12 bits indicating the manufacturer + 9 bits identifying the part + 3 bits indicating the revision) that must be sent to the host after reception of a continuous device-ID field as normal data for transmission. For details of the information that must be included in device-ID fields, contact NXP Semiconductors.

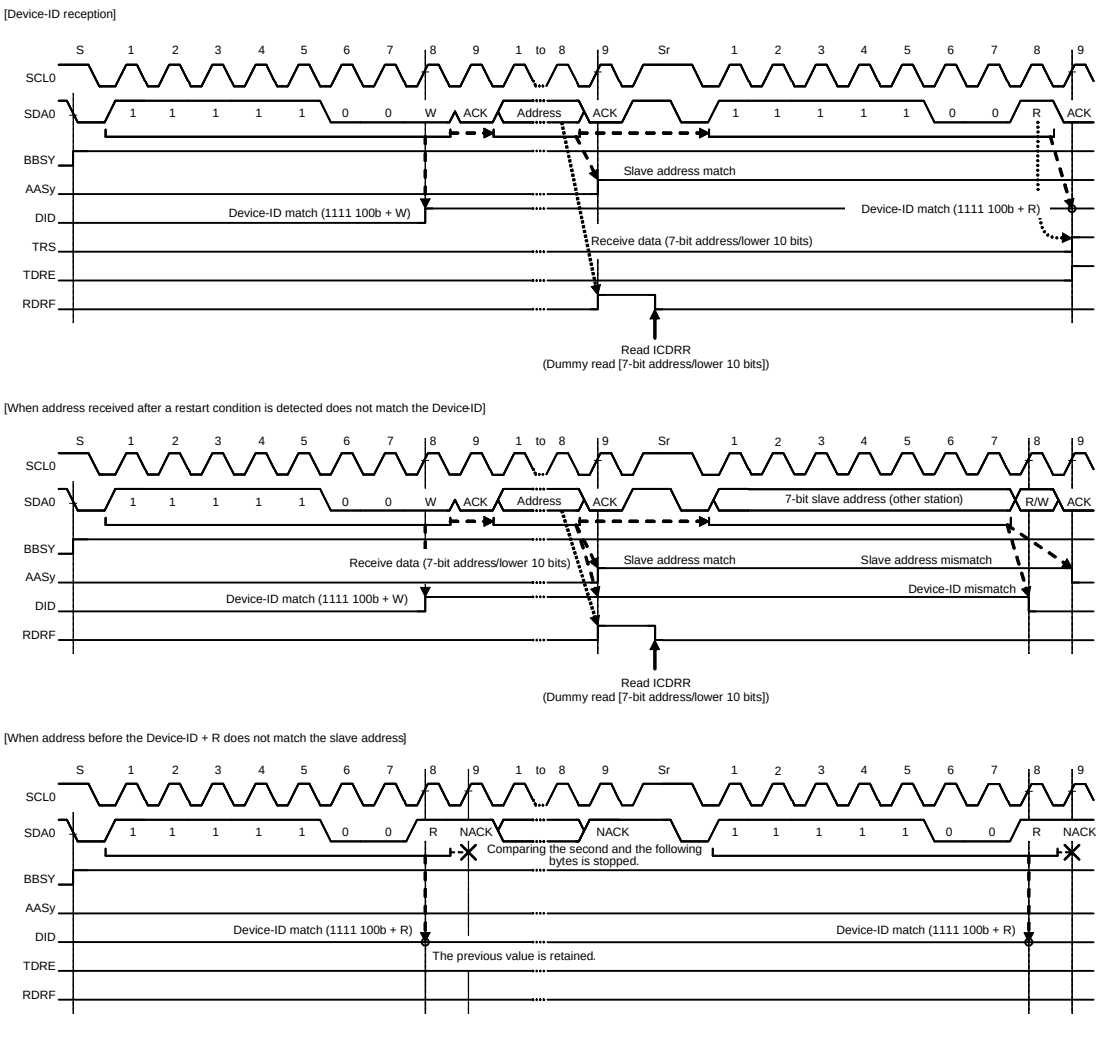



**Figure 27.28 AASy/DID Flag Set/Clear Timing during Reception of Device-ID**

### 27.7.4 Host Address Detection

The RIIC has a function to detect the host address while the SMBus is operating. When the HOAE bit in ICSER is set to 1 while the SMBS bit in ICMR3 is 1, the RIIC can detect the host address (0001 000b) in slave receive mode (MST and TRS bits = 00b in ICCR2).

When the RIIC detects the host address, the HOA flag in ICSR1 is set to 1 at the rising edge of the ninth SCL clock cycle, and at the same time, the RDRF flag in ICSR2 is set to 1 when the R/W# bit is 0 (Wr bit). This causes a receive data full interrupt (RXI) to be generated. The HOA flag is used to recognize that the host address was sent from the smart battery or other devices.

If the bit following the host address (0001 000b) is an Rd bit (R/W# bit = 1), the RIIC can also detect the host address. After the host address is detected, the RIIC operates in the same manner as normal slave operation.

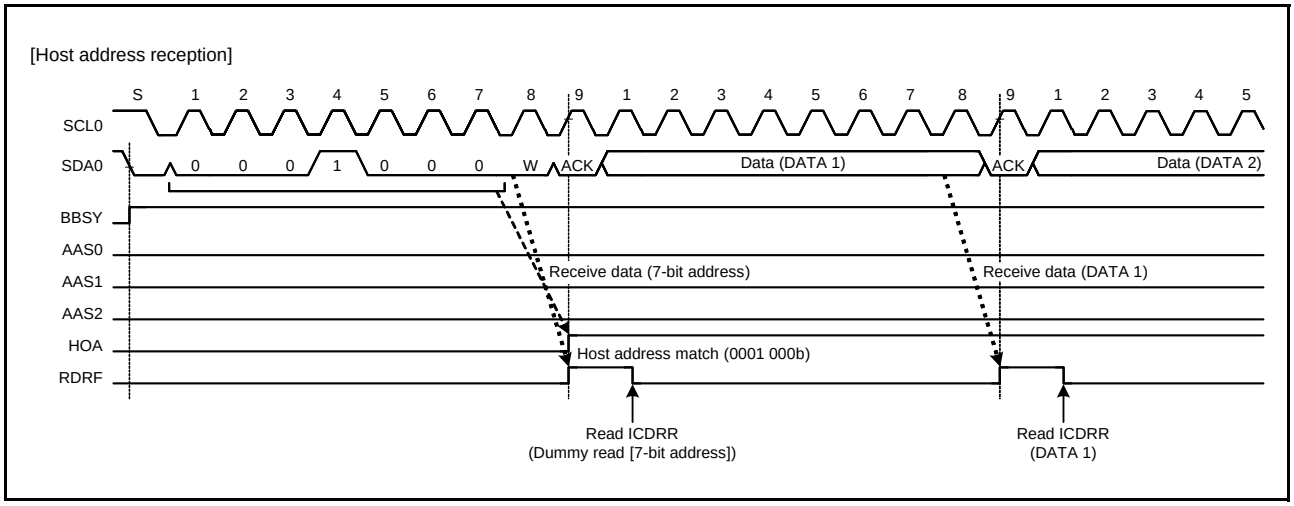



**Figure 27.29 HOA Flag Set Timing during Reception of Host Address**

## 27.8 Automatic Low-Hold Function for SCL

### 27.8.1 Function to Prevent Wrong Transmission of Transmit Data

If the shift register (ICDRS) is empty when data have not been written to the I2C bus transmit data register (ICDRT) with the RIIC in transmission mode (TRS bit = 1 in ICCR2), the SCL0 line is automatically held at the low level over the intervals shown below. This low-hold period is extended until data for transmission have been written, which prevents the unintended transmission of erroneous data.

Master transmit mode

- Low-level interval after a start condition or restart condition is issued
- Low-level interval between the ninth clock cycle of one transfer and the first clock cycle of the next

Slave transmit mode

- Low-level interval between the ninth clock cycle of one transfer and the first clock cycle of the next

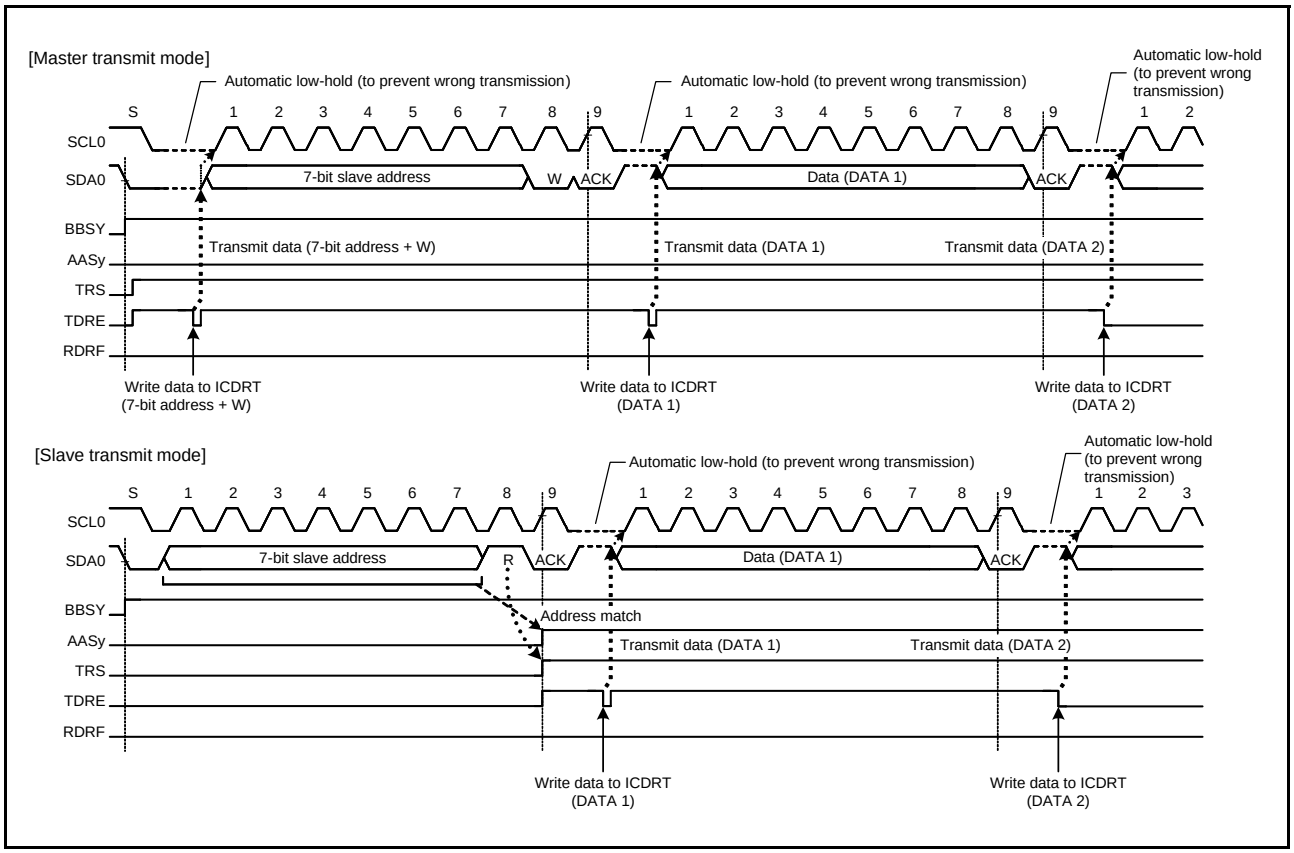



**Figure 27.30 Automatic Low-Hold Operation in Transmit Mode**

## 27.8.2 NACK Reception Transfer Suspension Function

The RIIC has a function to suspend transfer operation when NACK is received in transmit mode (TRS bit = 1 in ICCR2). This function is enabled when the NACKE bit in ICFER is set to 1 (transfer suspension enabled). If the next transmit data has already been written (TDRE flag = 0 in ICSR2) when NACK is received, next data transmission at the falling edge of the ninth SCL clock cycle is automatically suspended. This prevents the SDA0 line output level from being held low when the MSB of the next transmit data is 0.

If the transfer operation is suspended by this function (NACKF flag = 1 in ICSR2), transmit operation and receive operation are discontinued. To restore transmit/receive operation, be sure to clear the NACKF flag to 0. In master transmit mode, clear the NACKF flag to 0, issue a restart or stop condition, and then issue a start condition again.

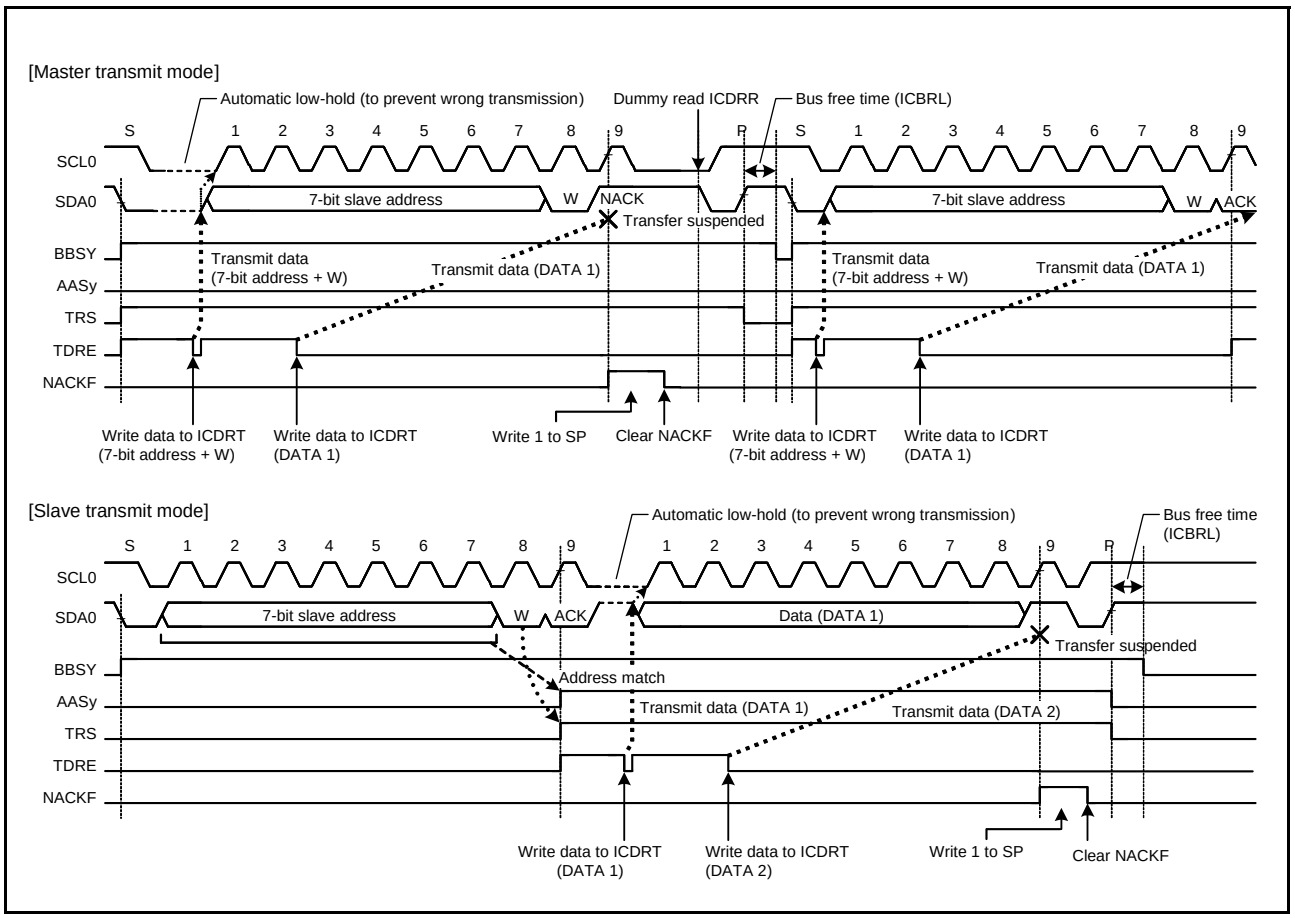



**Figure 27.31 Suspension of Data Transfer When NACK is Received (NACKE = 1)**

## 27.8.3 Function to Prevent Failure to Receive Data

If response processing is delayed when receive data (ICDRR) read is delayed for a period of one transfer frame or more with receive data full (RDRF flag = 1 in ICSR2) in receive mode (TRS = 0 in ICCR2), the RIIC holds the SCL0 line low automatically immediately before the next data is received to prevent failure to receive data.

This function to prevent failure to receive data using the automatic low-hold function is also enabled even if the read processing of the final receive data is delayed and, in the meantime, the RIIC's own slave address is designated after a stop condition is issued. This function does not disturb other communication because the RIIC does not hold the SCL0 line low when a mismatch with its own slave address occurs after a stop condition is issued.

Sections in which the SCL0 line is held low can be selected with a combination of the WAIT and RDRFS bits in ICMR3.

### (1) 1-Byte Receive Operation and Automatic Low-Hold Function Using the WAIT Bit

When the WAIT bit in ICMR3 is set to 1, the RIIC performs 1-byte receive operation using the WAIT bit function. Furthermore, when the ICMR3.RDRFS bit is 0, the RIIC automatically sends the ACKBT bit value in ICMR3 for the acknowledge bit in the period from the falling edge of the eighth SCL clock cycle to the falling edge of the ninth SCL clock cycle, and automatically holds the SCL0 line low at the falling edge of the ninth SCL clock cycle using the WAIT bit function. This low-hold is released by reading data from ICDRR, which enables bytewise receive operation.
The WAIT bit function is enabled for receive frames after a match with the RIIC's own slave address (including the general call address and host address) is obtained in master receive mode or slave receive mode.

### (2) 1-Byte Receive Operation (ACK/NACK Transmission Control) and Automatic Low-Hold Function Using the RDRFS Bit

When the RDRFS bit in ICMR3 is set to 1, the RIIC performs 1-byte receive operation using the RDRFS bit function. When the RDRFS bit is set to 1, the RDRF flag (receive data full) in ICSR2 is set to 1 at the rising edge of the eighth SCL clock cycle, and the SCL0 line is automatically held low at the falling edge of the eighth SCL clock cycle. This low-hold is released by writing a value to the ACKBT bit in ICMR3, but cannot be released by reading data from ICDRR, which enables receive operation by the ACK/NACK transmission control according to the data received in byte units.
The RDRFS bit function is enabled for receive frames after a match with the RIIC's own slave address (including the general call address and host address) is obtained in master receive mode or slave receive mode.

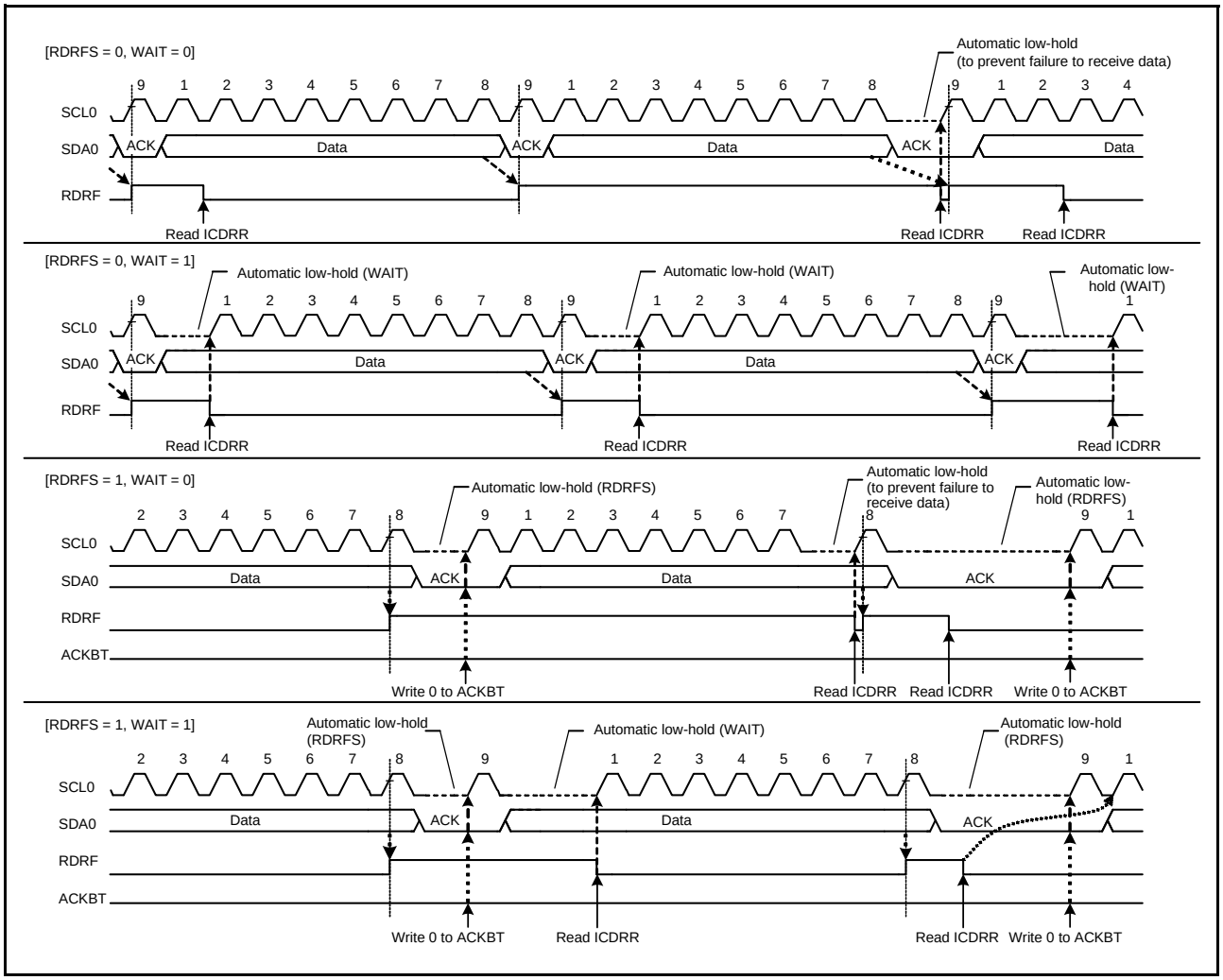



**Figure 27.32 Automatic Low-Hold Operation in Receive Mode (Using RDRFS and WAIT Bits)**

## 27.9 Arbitration-Lost Detection Functions

In addition to the normal arbitration-lost detection function defined by the I2C bus standard, the RIIC has functions to prevent double-issue of a start condition, to detect arbitration-lost during transmission of NACK, and to detect arbitration-lost in slave transmit mode.

### 27.9.1 Master Arbitration-Lost Detection (MALE Bit)

The RIIC drives the SDA0 line low to issue a start condition. However, if the SDA0 line has already been driven low by another master device issuing a start condition, the RIIC regards its own issuing of a start condition as an error and considers this a loss in arbitration, so priority is given to transfer by the other master device. Similarly, if a request to issue a start condition is made by setting the ST bit in ICCR2 to 1 while the bus is busy (BBSY flag = 1 in ICCR2), the RIIC regards this as a double-issuing-of-start-condition error and considers itself to have lost in arbitration, thus preventing a failure of transfer due to issuing of a start condition while transfer is in progress.
When a start condition is issued successfully, if the data for transmission including the address bits (i.e. the internal SDA output level) and the level on the SDA0 line do not match (the high output as the internal SDA output; i.e. the SDA0 pin is in the high-impedance state) and the low level is detected on the SDA0 line, the RIIC loses in arbitration.
After a loss in arbitration of mastership, the RIIC immediately enters slave receive mode. If a slave address (including the general call address) matches its own address at this time, the RIIC continues in slave operation.
A loss in arbitration of mastership is detected when the following conditions are met while the MALE bit in ICFER is 1 (master arbitration-lost detection enabled).
[Master arbitration-lost conditions]

- Non-matching of the internal level for output on SDA and the level on the SDA0 line after a start condition was issued by setting the ST bit in ICCR2 to 1 while the BBSY flag in ICCR2 was cleared to 0 (erroneous issuing of a start condition)
- Setting of the ST bit in ICCR2 to 1 (start condition double-issue error) while the BBSY flag is set to 1
- When the transmit data excluding acknowledge (internal SDA output level) does not match the level on the SDA0 line in master transmit mode (MST and TRS bits = 11b in ICCR2)

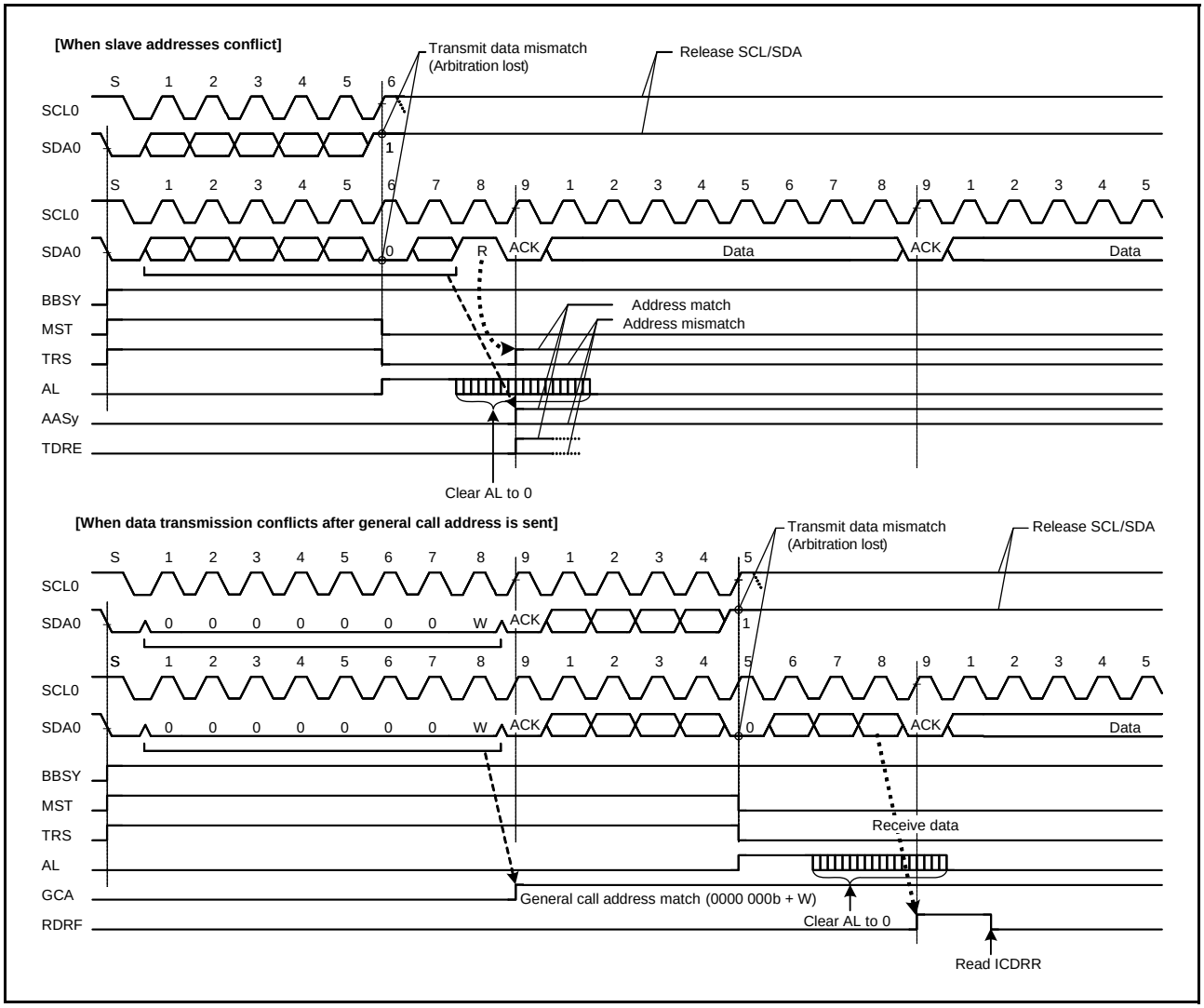



**Figure 27.33 Examples of Master Arbitration-Lost Detection (MALE = 1)**

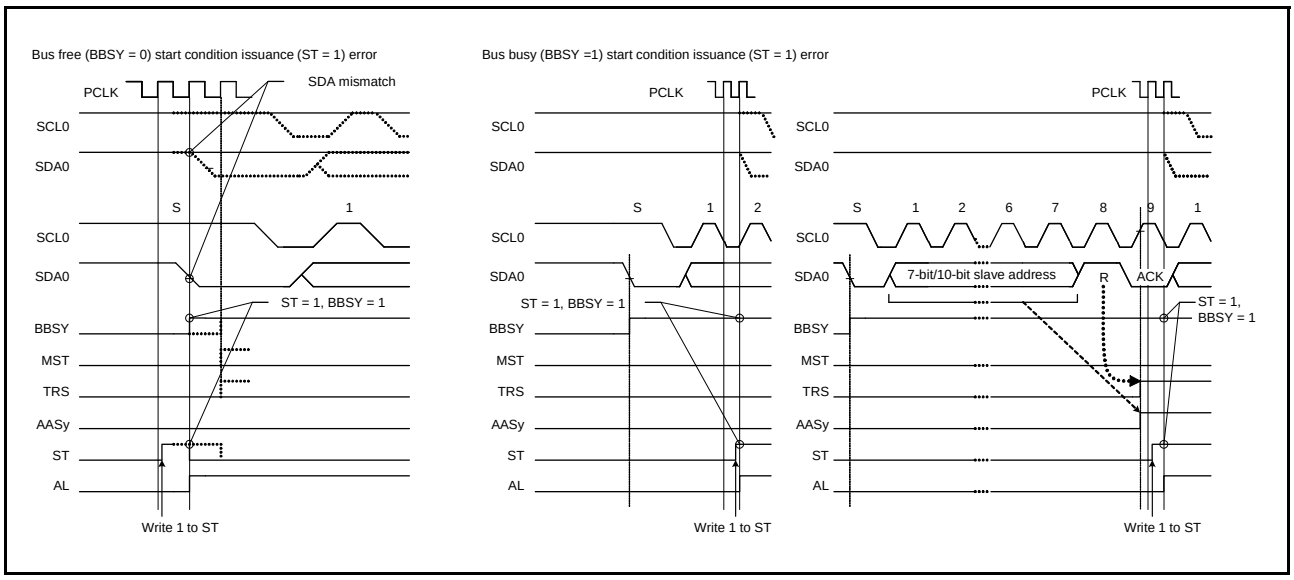



**Figure 27.34 Arbitration-Lost When a Start Condition is Issued (MALE = 1)**

## 27.9.2 Function to Detect Loss of Arbitration during NACK Transmission (NALE Bit)

The RIIC has a function to cause arbitration to be lost if the internal SDA output level does not match the level on the SDA0 line (the high output as the internal SDA output; i.e. the SDA0 pin is in the high-impedance state) and the low level is detected on the SDA0 line during transmission of NACK in receive mode. Arbitration is lost due to a conflict of NACK transmission and ACK transmission when two or more master devices receive data from the same slave device simultaneously in a multi-master system. Such conflict occurs when multiple master devices send/receive the same information through a single slave device. Figure 27.35 shows an example of arbitration-lost detection during transmission of NACK.

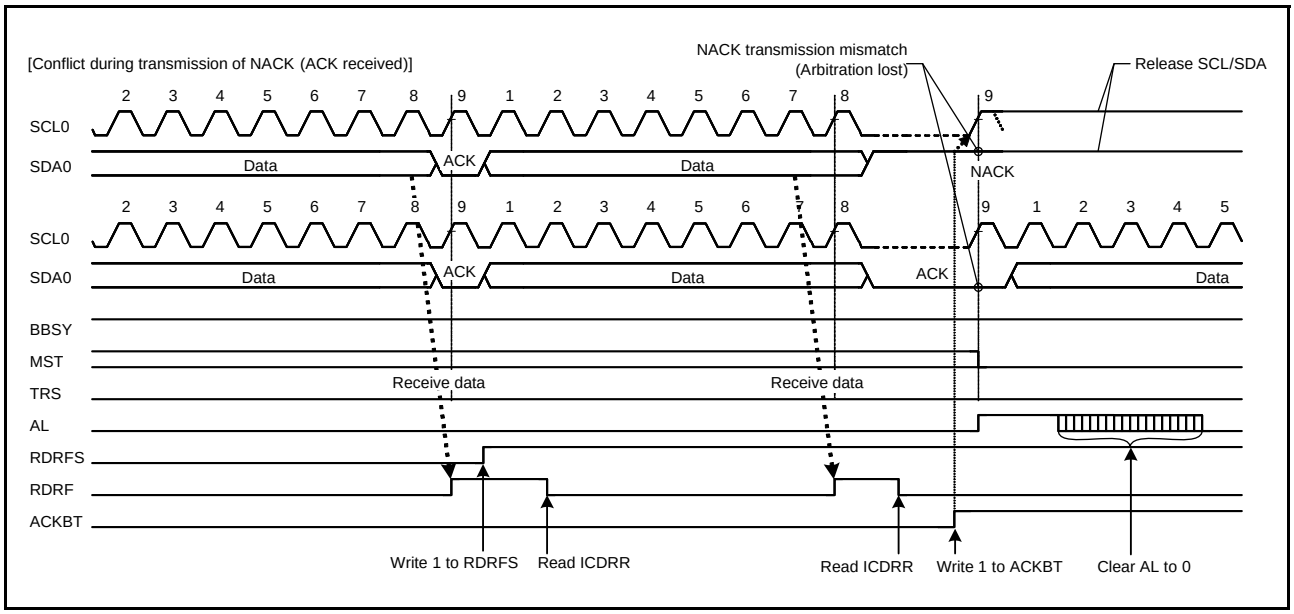



**Figure 27.35 Example of Arbitration-Lost Detection during Transmission of NACK (NALE = 1)**

The following explains arbitration-lost detection using an example where two master devices (master A and master B) and a single slave device are connected through the bus. In this example, master A receives two bytes of data from the slave device, and master B receives four bytes of data from the slave device.

If master A and master B access the slave device simultaneously, because the slave address is identical, arbitration is not lost in both master A and master B during access to the slave device. Therefore, both master A and master B recognize that they have obtained the bus mastership and operate as such. Here, master A sends NACK when it has received two final bytes of data from the slave device. Meanwhile, master B sends ACK because it has not received necessary four bytes of data. At this time, the NACK transmission from master A and the ACK transmission from master B conflict. In general, if a conflict like this occurs, master A cannot detect ACK transmitted by master B and issues a stop condition. Therefore, the issuance of the stop condition conflicts with the SCL clock output of master B, which disturbs communication.

When the RIIC receives ACK during transmission of NACK, it detects a defeat in conflict with other master devices and causes arbitration to be lost.

If arbitration is lost during transmission of NACK, the RIIC immediately cancels the slave match condition and enters slave receive mode. This prevents a stop condition from being issued, preventing a communication failure on the bus.

Similarly, in the ARP command processing of SMBus, the function to detect loss of arbitration during transmission of NACK is also available for eliminating the extra clock cycle processing (such as FFh transmission processing) necessary if the UDID (Unique Device Identifier) of assign address does not match in the Get UDID (general) processing after the Assign address command.

The RIIC detects arbitration-lost during transmission of NACK when the following condition is met with the NALE bit in ICFER set to 1 (arbitration-lost detection during NACK transmission enabled).

[Condition for arbitration-lost during NACK transmission]

- When the internal SDA output level does not match the SDA0 line (ACK is received) during transmission of NACK (ACKBT bit = 1 in ICMR3)

### 27.9.3 Slave Arbitration-Lost Detection (SALE Bit)

The RIIC has a function to cause arbitration to be lost if the data for transmission (i.e. the internal SDA output level) and the level on the SDA0 line do not match (the high output as the internal SDA output; i.e. the SDA0 pin is in the high-impedance state) and the low level is detected on the SDA0 line in slave transmit mode. This arbitration-lost detection function is mainly used when transmitting a UDID (Unique Device Identifier) over an SMBus.
When it loses slave arbitration, the RIIC is immediately released from the slave-matched state and enters slave receive mode. This function can detect conflicts of data during transmission of UDIDs over an SMBus and eliminates subsequent redundant processing (processing for the transmission of FFh).
The RIIC detects slave arbitration-lost when the following condition is met with the SALE bit in ICFER set to 1 (slave arbitration-lost detection enabled).

[Condition for slave arbitration-lost]

- When transmit data excluding acknowledge (internal SDA output level) does not match the SDA0 line in slave transmit mode (MST and TRS bits = 01b in ICCR2)

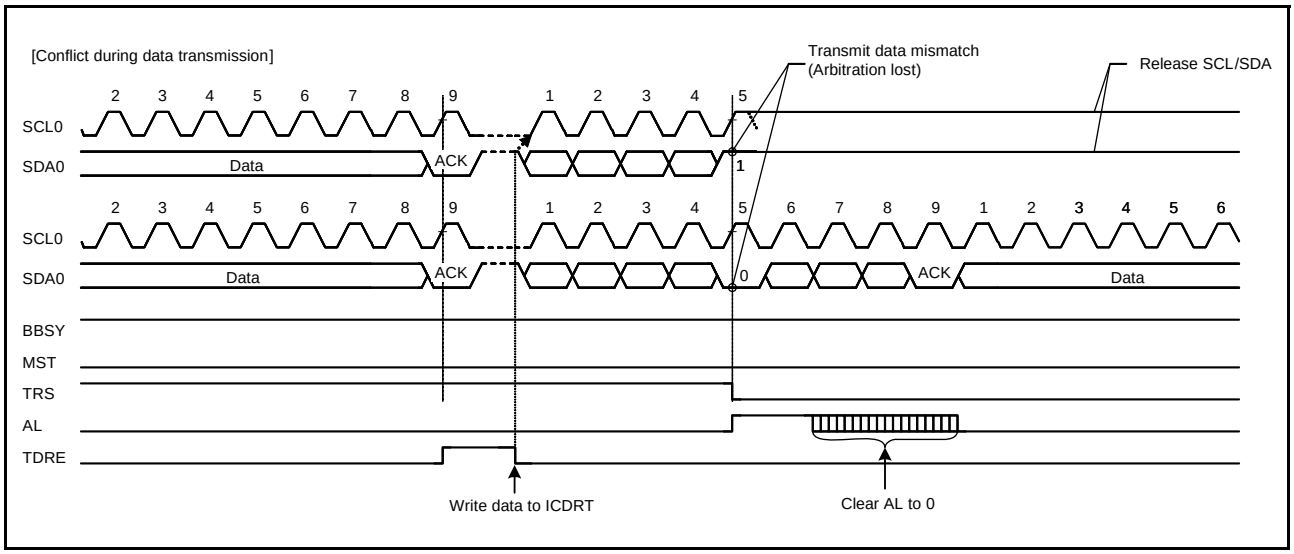



**Figure 27.36 Example of Slave Arbitration-Lost Detection (SALE = 1)**

## 27.10 Start Condition/Restart Condition/Stop Condition Issuing Function

### 27.10.1 Issuing a Start Condition

The RIIC issues a start condition when the ST bit in ICCR2 is set to 1.
When the ST bit is set to 1, a start condition issuance request is made and the RIIC issues a start condition when the BBSY flag in ICCR2 is 0 (bus free state). When a start condition is issued normally, the RIIC automatically shifts to the master transmit mode.
A start condition is issued in the following sequence.

[Start condition issuance]
(1) Drive the SDA0 line low (high level to low level).
(2) Ensure the time set in ICBRH and the start condition hold time.
(3) Drive the SCL0 line low (high level to low level).
(4) Detect low level of the SCL0 line and ensure the low-level period of SCL0 line set in ICBRL.

### 27.10.2 Issuing a Restart Condition

The RIIC issues a restart condition when the RS bit in ICCR2 is set to 1.
When the RS bit is set to 1, a restart condition issuance request is made and the RIIC issues a restart condition when the BBSY flag in ICCR2 is 1 (bus busy state) and the MST bit in ICCR2 is 1 (master mode).
A restart condition is issued in the following sequence.

[Restart condition issuance]
(1) Release the SDA0 line.
(2) Ensure the low-level period of SCL0 line set in ICBRL.
(3) Release the SCL0 line (low level to high level).
(4) Detect a high level of the SCL0 line and ensure the time set in ICBRL and the restart condition setup time.
(5) Drive the SDA0 line low (high level to low level).
(6) Ensure the time set in ICBRH and the restart condition hold time.
(7) Drive the SCL0 line low (high level to low level).
(8) Detect a low level of the SCL0 line and ensure the low-level period of SCL0 line set in ICBRL.

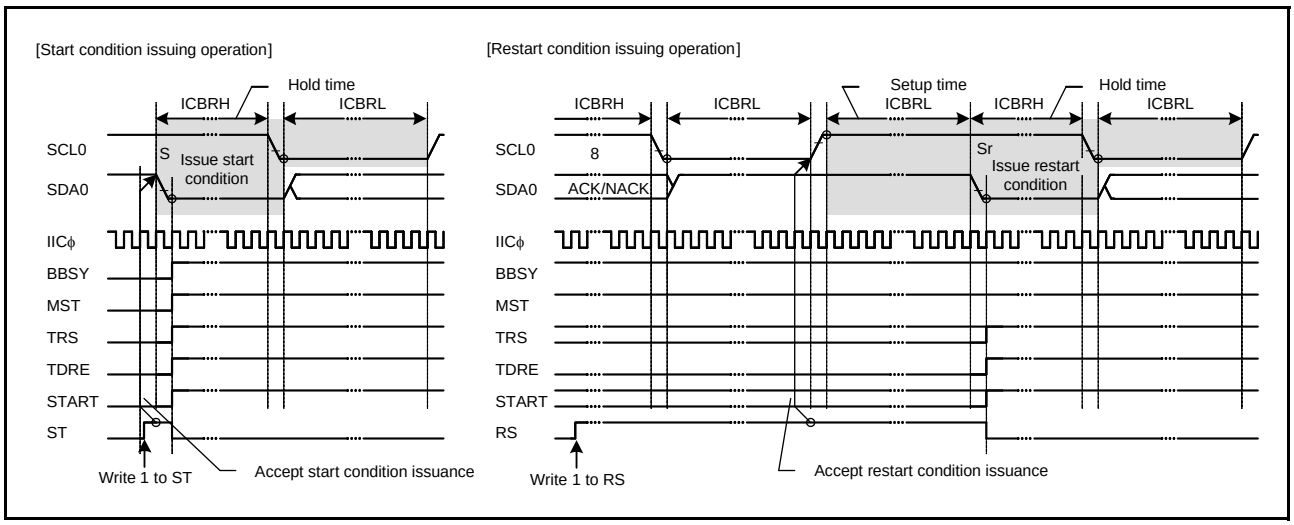



**Figure 27.37 Start Condition/Restart Condition Issue Timing (ST and RS Bits)**

## 27.10.3 Issuing a Stop Condition

The RIIC issues a stop condition when the SP bit in ICCR2 is set to 1.
When the SP bit is set to 1, a stop condition issuance request is made and the RIIC issues a stop condition when the BBSY flag in ICCR2 is 1 (bus busy state) and the MST bit in ICCR2 is 1 (master mode).
A stop condition is issued in the following sequence.

[Stop condition issuance]

- Drive the SDA0 line low (high level to low level).
- Ensure the low-level period of SCL0 line set in ICBRL.
- Release the SCL0 line (low level to high level).
- Detect a high level of the SCL0 line and ensure the time set in ICBRH and the stop condition setup time.
- Release the SDA0 line (low level to high level).
- Ensure the time set in ICBRL and the bus free time.
- Clear the BBSY flag to 0 (to release the bus mastership).

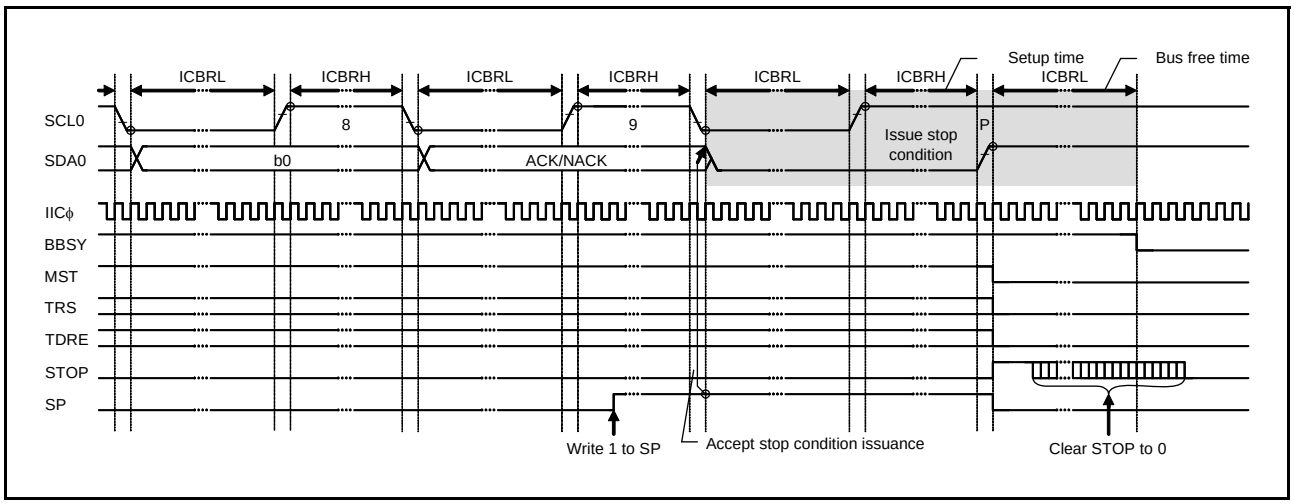



**Figure 27.38 Stop Condition Issue Timing (SP Bit)**

## 27.11 Bus Hanging

If the clock signals from the master and slave devices go out of synchronization due to noise or other factors, the I2C bus might hang with a fixed level on the SCL0 line and/or SDA0 line.
As measures against the bus hanging, the RIIC has a timeout function to detect hanging by monitoring the SCL0 line, a function for the output of an extra SCL clock cycle to release the bus from a hung state due to clock signals being out of synchronization, the RIIC reset function, and internal reset function.
By checking the SCLO, SDAO, SCLI, and SDAI bits in ICCR1, it is possible to see whether the RIIC or its partner in communications is placing the low level on the SCL0 or SDA0 lines.

### 27.11.1 Timeout Function

The RIIC includes a timeout function for detecting when the SCL0 line has been stuck longer than the predetermined time. The RIIC can detect an abnormal bus state by monitoring that the SCL0 line is stuck low or high for a predetermined time.
The timeout function monitors the SCL0 line state and counts the low-level period or high-level period using the internal counter. The timeout function resets the internal counter each time the SCL0 line changes (rising or falling), but continues to count unless the SCL0 line changes. If the internal counter overflows due to no SCL0 line change, the RIIC can detect the timeout and report the bus hung state.
This timeout function is enabled when the ICFER.TMOE bit is 1. It detects a hung state that the SCL0 line is stuck low or high during the following conditions:

- The bus is busy (ICCR2.BBSY flag is 1) in master mode (ICCR2.MST bit is 1).
- The RIIC's own slave address is detected (ICSR1 register is not 00h) and the bus is busy (ICCR2.BBSY flag is 1) in slave mode (ICCR2.MST bit is 0).
- The bus is free (ICCR2.BBSY flag is 0) while generation of a START condition is requested (ICCR2.ST bit is 1).

The internal counter of the timeout function works using the internal reference clock (IICϕ) set by the CKS[2:0] bits in ICMR1 as a count source. It functions as a 16-bit counter when long mode is selected (TMOS bit = 0 in ICMR2) or a 14-bit counter when short mode is selected (TMOS bit = 1).
The SCL0 line level (low/high or both levels) during which this counter is activated can be selected by the setting of the TMOH and TMOL bits in ICMR2. If both TMOL and TMOH bits are cleared to 0, the internal counter does not work.

Note: • When using the timeout detection function, refer to section 27.2.4, I2C Bus Mode Register 2 (ICMR2), section 27.2.18, Timeout Internal Counter (TMOCNT), and section 27.3.2, Initial Settings.

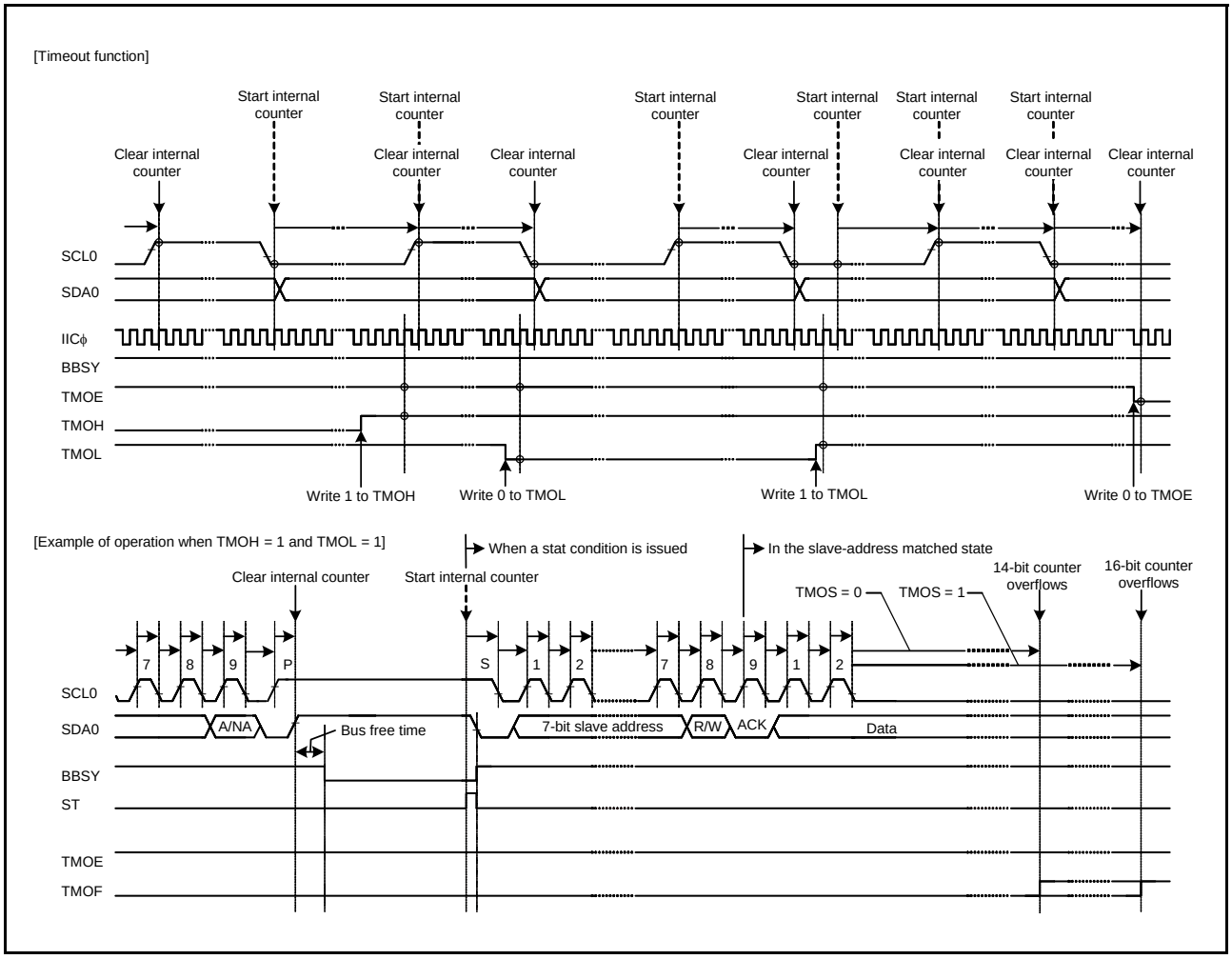



**Figure 27.39 Timeout Function (TMOE, TMOS, TMOH, and TMOL Bits)**

## 27.11.2 Extra SCL Clock Cycle Output Function

In master mode, the RIIC module has a facility for the output of extra SCL clock cycles to release the SDA0 line of the slave device from being held at the low level due to the master being out of synchronization with the slave device.
This function is mainly used in master mode to release the SDA0 line of the slave device from the state of being fixed to the low level by including extra cycles of SCL output from the RIIC with single cycles of the SCL clock as the unit in the case of a bus error where the RIIC cannot issue a stop condition because the slave device is holding the SDA0 line at the low level. Do not use this facility in normal situations. Using it when communications are proceeding correctly will lead to malfunctions.
When the CLO bit in ICCR1 is set to 1 in master mode, a single cycle of the SCL clock at the frequency corresponding to the transfer rate settings (settings of the CKS[2:0] bits in ICMR1, and of the ICBRH and ICBRL registers) is output as an extra clock cycle. After output of this single cycle of the SCL clock, the CLO bit is automatically cleared to 0. Therefore, further extra clock cycles can be output consecutively by the software program writing 1 to the CLO bit after having read CLO = 0.
When the RIIC module is in master mode and the slave device is holding the SDA0 line at the low level because synchronization with the slave device has been lost due to the effects of noise, etc., the output of a stop condition is not possible. The facility for output of an extra cycle of the SCL clock can be used to output extra cycles of SCL one by one to make the slave device release the SDA0 line from being held at the low level, thus recovering the bus from an unusable state. Release of the SDA0 line by the slave device can be monitored by reading the SDAI bit in ICCR1. After confirming release of the SDA0 line by the slave device, complete communications by reissuing the stop condition.
Use this facility with the MALE bit (master arbitration-lost detection disabled) in ICFER cleared to 0. If the MALE bit is set to 1 (master arbitration-lost detection enabled), arbitration is lost when the value of the SDAO bit in ICCR1 does not match the state of the SDA0 line, so take care on this point.

[Output conditions for using the CLO bit in ICCR1]

- When the bus is free (BBSY flag in ICCR2 = 0) or in master mode (MST bit = 1 and BBSY flag = 1 in ICCR2)
- When the communication device does not hold the SCL0 line low

Figure 27.40 shows the operation timing of the extra SCL clock cycle output function (CLO bit).

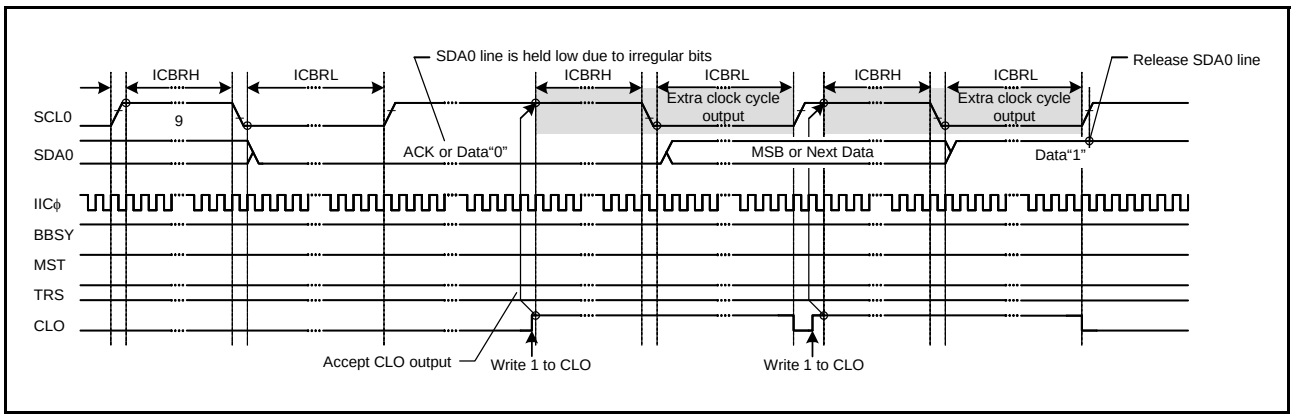



**Figure 27.40 Extra SCL Clock Cycle Output Function (CLO Bit)**

### 27.11.3 RIIC Reset and Internal Reset

The RIIC module incorporates a function for resetting itself. There are two types of reset. One is referred to as an RIIC reset; this initializes all registers including the BBSY flag in ICCR2. The other is referred to as an internal reset; this releases the RIIC from the slave-address matched state and initializes the internal counter while retaining other settings. After issuing a reset, be sure to clear the IICRST bit in ICCR1 to 0.

Both types of reset are effective for release from bus-hung states since both restore the output state of the SCL0 and SDA0 pins to the high impedance state.

Issuing a reset during slave operation may lead to a loss of synchronization between the master device clock and the slave device clock, so avoided this where possible. Note that monitoring of the bus state, such as for the presence of a start condition, is not possible during an RIIC reset (ICE and IICRST bits = 01b in ICCR1).

For a detailed description of the RIIC and internal resets, refer to section 27.14, Resets and Register and Function States When Issuing Each Condition.

## 27.12 SMBus Operation

The RIIC is available for data communication conforming to the SMBus (Version 2.0). To perform SMBus communication, set the SMBS bit in ICMR3 to 1. To use the transfer rate within a range of 10 kbps to 100 kbps of the SMBus standard, set the CKS[2:0] bits in ICMR1, ICBRH, and ICBRL. In addition, determine the values of the DLCS bit in ICMR2 and the SDDL[2:0] bits in ICMR2 to meet the data hold time specification of 300 ns or more. If the RIIC is used only as a slave device, the transfer rate setting is not necessary. When the RIIC is used only as a slave device, the transfer rate setting is not necessary, whereas the ICBRL needs to be set to a value longer than the data setup time (250 ns).

For the SMBus device default address (1100 001b), use one of the slave address registers L0 to L2 (SARL0, SARL1, and SARL2), and set the corresponding FS bit (7-bit/10-bit address format select) in SARUy (y = 0 to 2) to 0 (7-bit address format).

When transmitting the UDID (Unique Device Identifier), set the SALE bit in ICFER to 1 to enable the slave arbitration-lost detection function.

### 27.12.1 SMBus Timeout Measurement

#### (1) Measuring timeout of slave device

The following period (timeout interval: $T_{LOW:\ SEXT}$) must be measured for slave devices in SMBus communication.

- From start condition to stop condition

To measure timeout for slave devices, measure the period from start condition detection to stop condition detection with the MTU timer using a start condition detection interrupt (STI) and stop condition detection interrupt (SPI) of the RIIC. The measured timeout period must be within the total clock low-level period [slave device] $T_{LOW:\ SEXT}$: 25 ms (max.) of the SMBus standard.

If the time measured with the MTU exceeds the clock low-level detection timeout $T_{TIMEOUT}$: 25 ms (min.) of the SMBus standard, the slave device must release the bus by writing 1 to the IICRST bit in ICCR1 to issue an internal reset of the RIIC. When an internal reset is issued, the RIIC stops driving the bus for the SCL0 pin and SDA0 pin and make the SCL0/SDA0 pin outputs high impedance, which releases the bus.

#### (2) Measuring timeout of master device

The following periods (timeout interval: $T_{LOW:\ MEXT}$) must be measured for master devices in SMBus communication.

- From start condition to acknowledge bit
- Between acknowledge bits
- From acknowledge bit to stop condition

To measure timeout for master devices, measure these periods with the MTU timer using a start condition detection interrupt (STI), stop condition detection interrupt (SPI), and transmit end interrupt (TEI) or receive data full interrupt (RXI) of the RIIC. The measured timeout period must be within the total clock low-level extended period (master device) $T_{LOW:\ MEXT}$: 10 ms (max.) of the SMBus standard, and the total of all $T_{LOW:\ MEXT}$ from start condition to stop condition must be within $T_{LOW:\ SEXT}$: 25 ms (max.).

For the ACK receive timing (rising edge of the ninth SCL clock cycle), monitor the TEND flag in ICSR2 in master transmit mode (master transmitter) and the RDRF flag in ICSR2 in master receive mode (master receiver). For this reason, perform bytewise transmit operation in master transmit mode, and hold the RDRFS bit in ICMR3 0 until the byte just before reception of the final byte in master receive mode. While the RDRFS bit is 0, the RDRF flag is set to 1 at the rising edge of the ninth SCL clock cycle.

If the period measured with the MTU exceeds the total clock low-level extended period (master device) $T_{LOW:\ MEXT}$: 10 ms (max.) of the SMBus standard or the total of measured periods exceeds the clock low-level detection timeout $T_{TIMEOUT}$: 25 ms (min.) of the SMBus standard, the master device must stop the transaction by issuing a stop condition. In master transmit mode, immediately stop the transmit operation (writing data to ICDRT).

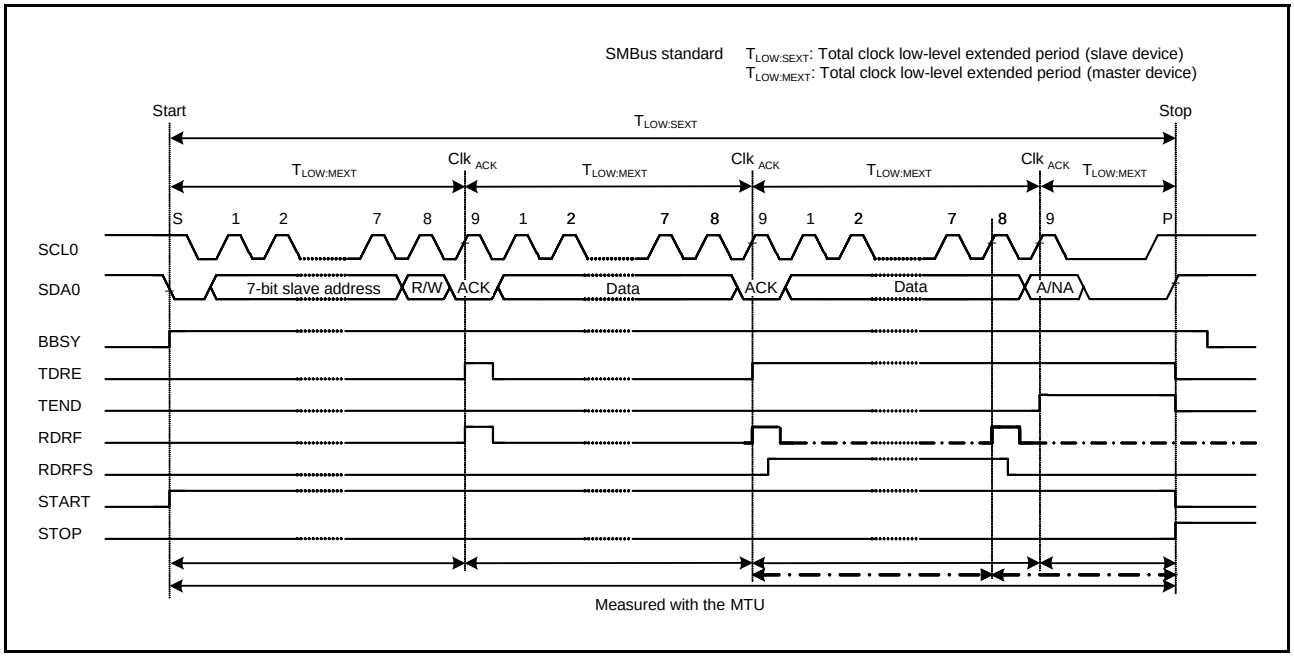



**Figure 27.41 SMBus Timeout Measurement**

## 27.12.2 Packet Error Code (PEC)

This MCU incorporates a CRC calculator. The CRC calculator enables transmission of a packet error code (PEC) or checking the received data of the SMBus in data communication of the RIIC. For the CRC generating polynomials of the CRC calculator, refer to section 29, CRC Calculator (CRC).

The PEC data in master transmit mode can be generated by writing all transmit data to the CRC data input register (CRCDIR) in the CRC calculator.

The PEC data in master receive mode can be checked by writing all receive data to CRCDIR in the CRC calculator and comparing the obtained value in the CRC data output register (CRCDOR) with the received PEC data.

To send ACK or NACK according to the match or mismatch result when the final byte is received as a result of the PEC code check, set the RDRFS bit in ICMR3 to 1 before the rising edge of the eighth SCL clock cycle during reception of the final byte, and hold the SCL0 line low at the falling edge of the eighth clock cycle.

## 27.12.3 SMBus Host Notification Protocol (Notify ARP Master Command)

In communications over an SMBus, a slave device can temporarily act as a master device to notify the SMBus host (or ARP master) of (or request the SMBus host for) its own slave address or to request its own slave address from the SMBus host.

For a product of the this MCU to operate as an SMBus host (or ARP master), the host address (0001 000b) sent from the slave device must be detected as a slave address, so the RIIC has a function for detecting the host address. To detect the host address as a slave address, set the SMBS bit in ICMR3 and the HOAE bit in ICSER to 1. Operation after the host address has been detected is the same as normal slave operation.

## 27.13 Interrupt Request

The RIIC issues four types of interrupt request: transfer error or event generation (arbitration-lost, NACK detection, timeout detection, start condition detection, and stop condition detection), receive data full, transmit data empty, and transmit end.

Table 27.7 lists details of the several interrupt requests. The receive data full and transmit data empty are both capable of launching data transfer by the DTC.

**Table 27.7 Interrupt Sources**

| Symbol | Interrupt Source | Interrupt Flag | DTC Launching | Priority | Interrupt Condition |
|---|---|---|---|---|---|
| EEI | Transfer Error/ Event Generation | AL | Not possible | High | AL = 1 • ALIE = 1 |
| | | NACKF | | | NACKF = 1 • NAKIE = 1 |
| | | TMOF | | | TMOF = 1 • TMOIE = 1 |
| | | START | | | START = 1 • STIE = 1 |
| | | STOP | | | STOP = 1 • SPIE = 1 |
| RXI*2 | Receive Data Full | — | Possible | | RDRF = 1 • RIE = 1 |
| TXI*1 | Transmit Data Empty | — | Possible | | TDRE = 1 • TIE = 1 |
| TEI*3 | Transmit End | TEND | Not possible | Low | TEND = 1 • TEIE = 1 |

Note: • There is a delay time between the execution of a write instruction for a peripheral module by the CPU and actual writing to the module. Thus, when an interrupt flag has been cleared or masked, read the relevant flag again to check whether clearing or masking has been completed, and then return from interrupt handling. Returning from interrupt handling without checking that writing to the module has been completed creates a possibility of repeated processing of the same interrupt.

Note 1. Since TXI is an edge-detected interrupted, it does not require clearing. Furthermore, the TDRE flag in ICSR2 (a condition for TXI) is automatically cleared to 0 when data for transmission are written to ICDRT or a stop condition is detected (STOP flag = 1 in ICSR2).

Note 2. Since RXI is an edge-detected interrupted, it does not require clearing. Furthermore, the RDRF flag in ICSR2 (a condition for RXI) is automatically cleared to 0 when data are read from ICDRR.

Note 3. When using the TEI interrupt, clear the TEND flag in ICSR2 in the TEI interrupt handling.
Note that the TEND flag in ICSR2 is automatically cleared to 0 when data for transmission are written to DRT or a stop condition is detected (STOP flag = 1 in ICSR2).

Clear or mask the each flag during interrupt handling.

### 27.13.1 Buffer Operation for TXI and RXI Interrupts

If the conditions for generating a TXI and RXI interrupt are satisfied while the corresponding IR flag is 1, the interrupt request is not output for the ICU but retained internally (the capacity for internal retention is one request per source). An interrupt request that was being retained within the ICU is output when the value of the ICU.IRn.IR flag becomes 0. Internally retained interrupt requests are automatically cleared under normal conditions of usage.

Internally retained interrupt requests can also be cleared by writing 0 to the interrupt enable bit within the given peripheral module.

## 27.14 Resets and Register and Function States When Issuing Each Condition

The RIIC has reset, RIIC reset, and internal reset functions. Table 27.8 lists the resets and register and function states when issuing each condition.

**Table 27.8 Resets and Register and Function States When Issuing Each Condition**

| | | Chip Reset | RIIC Reset (ICE = 0, IICRST = 1) | Internal Reset (ICE = 1, IICRST = 1) | Start Condition/ Restart Condition Detection | Stop Condition Detection |
|---|---|---|---|---|---|---|
| ICCR1 | ICE, IICRST | At a reset | Retained | Retained | Retained | Retained |
| | SCLO, SDAO | | At a reset | At a reset | | |
| | Others | | | Retained | | |
| ICCR2 | BBSY | At a reset | At a reset | Retained | Retained | Retained |
| | ST | | | At a reset | At a reset | Retained |
| | Others | | | | | At a reset |
| ICMR1 | BC[2:0] | At a reset | At a reset | At a reset | At a reset | Retained |
| | Others | | | Retained | Retained | |
| ICMR2 | | At a reset | At a reset | Retained | Retained | Retained |
| ICMR3 | | At a reset | At a reset | Retained | Retained | Retained |
| ICFER | | At a reset | At a reset | Retained | Retained | Retained |
| ICSER | | At a reset | At a reset | Retained | Retained | Retained |
| ICIER | | At a reset | At a reset | Retained | Retained | Retained |
| ICSR1 | | At a reset | At a reset | At a reset | Retained | At a reset |
| ICSR2 | TDRE, TEND | At a reset | At a reset | At a reset | Retained | At a reset |
| | START | | | | Retained | |
| | STOP | | | | Retained | Retained |
| | Others | | | | Retained | Retained |
| SARL0, SARL1, SARL2 SARU0, SARU1, SARU2 | | At a reset | At a reset | Retained | Retained | Retained |
| ICBRH, ICBRL | | At a reset | At a reset | Retained | Retained | Retained |
| ICDRT | | At a reset | At a reset | Retained | Retained | Retained |
| ICDRR | | At a reset | At a reset | Retained | Retained | Retained |
| ICDRS | | At a reset | At a reset | At a reset | Retained | Retained |
| Timeout function | | At a reset | At a reset | Operation | Operation | Operation |
| Bus free time measurement | | At a reset | At a reset | Operation | Operation | Operation |

## 27.15 Event Link Output

The RIIC0 handles event output for the event link controller (ELC) corresponding to the following sources.

(1) Transfer error event

When a transfer error event occurs, the corresponding event signal can be output for another module via the ELC.

(2) Received-data full

When a received data register becomes full, the corresponding event signal can be output for another module via the ELC.

(3) Transmission-data empty

When a transmission data register becomes empty, the corresponding event signal can be output for another module via the ELC.

(4) Transmission-completed

On completion of transfer, the corresponding event signal can be output for another module via the ELC.

### 27.15.1 Interrupt Handling and Event Linking

The RIIC module produces four kinds of interrupt: transfer-error (arbitration-lost detection, detection of NACK, detection of timeout, or detection of a stop condition) event, received data full, transmission data empty, and transmission completed interrupts detection of a start condition. Each of these has an enable bit to control enabling and disabling of the interrupt signal. An interrupt-request signal is output for the CPU when an interrupt-source condition is satisfied while the setting of the corresponding enable bit is “enabled”.

The corresponding event link output signals are sent to other modules as event signals via the ELC when the interrupt-source conditions are satisfied, regardless of the settings of the interrupt enable bits.

For details on interrupt sources, see Table 27.7.

## 27.16 Usage Notes

### 27.16.1 Setting Module Stop Function

Module stop state can be entered or canceled using module stop control register B (MSTPCRB). The initial setting is for operation of the RIIC to be stopped. RIIC register access is enabled by clearing module stop state.
For details on module stop control registers B, refer to section 11, Low Power Consumption.

### 27.16.2 Points to Note on Starting Transfer

If the IR flag corresponding to the RIIC interrupt is 1 when transfer is started (ICCR1.ICE bit = 1), follow the procedure below to clear interrupts before enabling operations. Starting transfer with the IR flag set to 1 while the ICCR1.ICE bit is 1 leads to an interrupt request being internally retained after transfer starts, and this can lead to unanticipated behavior of the IR flag.

1. Confirm that the ICCR1.ICE bit is 0.
2. Clear the relevant interrupt enable bits (ICIER.TIE, etc.) on the peripheral function side to 0.
3. Read the relevant interrupt enable bits (ICIER.TIE, etc.) on the peripheral function side and confirm that its value is 0.
4. Clear the IR flag to 0.

# 28. Serial Peripheral Interface (RSPI)

## 28.1 Overview

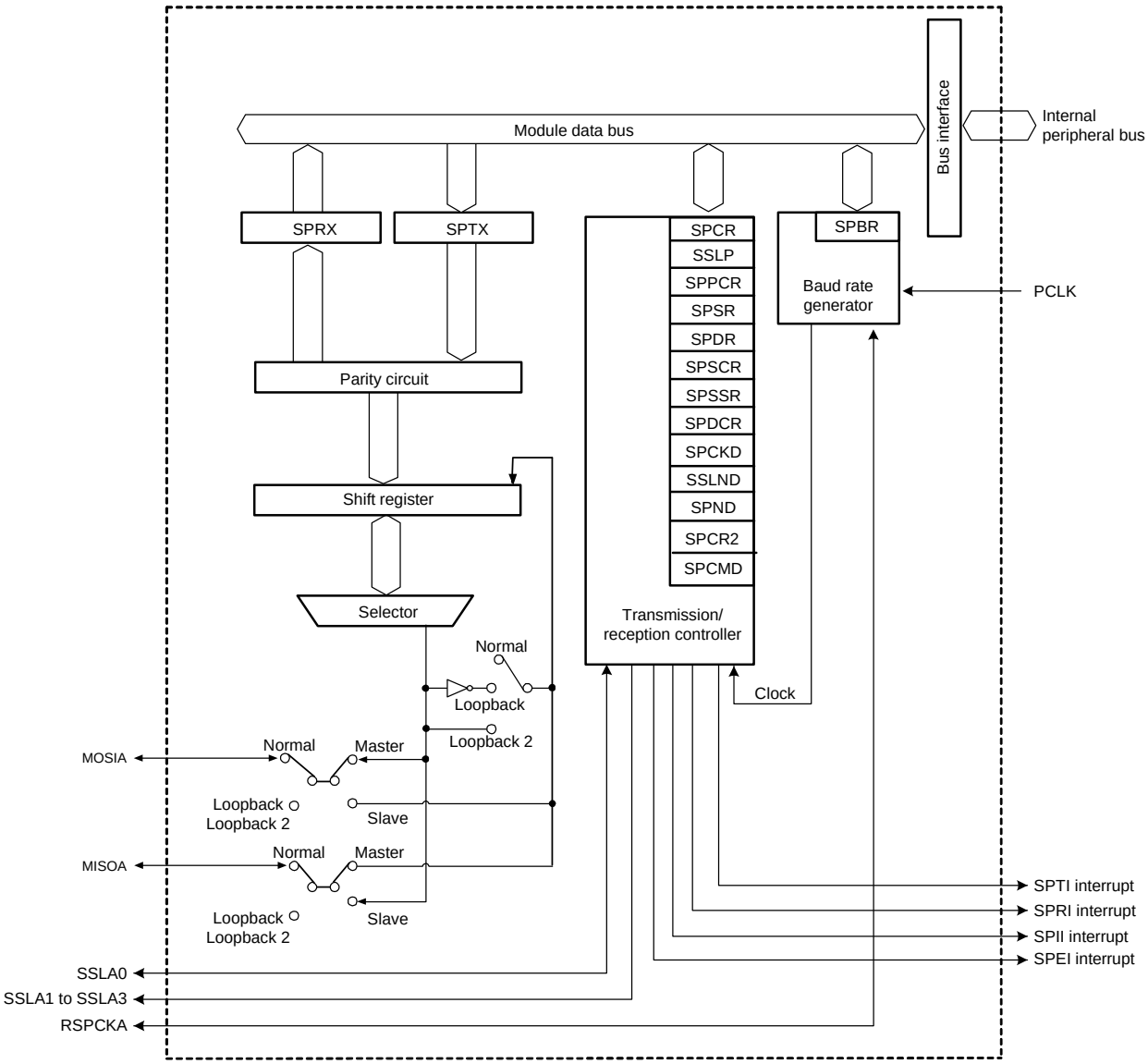
This MCU includes one channel of Serial Peripheral Interface (RSPI).
The RSPI channels are capable of high-speed, full-duplex synchronous serial communications with multiple processors and peripheral devices.
Table 28.1 lists the specifications of the RSPI, and Figure 28.1 shows a block diagram of the RSPI.
In this chapter, a lower-case letter m in RSPI command register m (SPCMDm) indicates a value from 0 to 7.

**Table 28.1 RSPI Specifications (1/2)**

| Item | Description |
|---|---|
| Number of channels | One channel |
| RSPI transfer functions | • Use of MOSI (master out/slave in), MISO (master in/slave out), SSL (slave select), and RSPCK (RSPI clock) signals allows serial communications through SPI operation (four-wire method) or clock synchronous operation (three-wire method).<br>• Transmit-only operation is available.<br>• Communication mode: Full-duplex or transmit-only can be selected.<br>• Switching of the polarity of RSPCK<br>• Switching of the phase of RSPCK |
| Data format | • MSB first/LSB first selectable<br>• Transfer bit length is selectable as 8, 9, 10, 11, 12, 13, 14, 15, 16, 20, 24, or 32 bits.<br>• 128-bit transmit/receive buffers<br>• Up to four frames can be transferred in one round of transmission/reception (each frame consisting of up to 32 bits). |
| Bit rate | • In master mode, the on-chip baud rate generator generates RSPCK by frequency-dividing PCLK (the division ratio ranges from divided by 2 to divided by 4096).<br>• In slave mode, the minimum PCLK clock divided by 8 can be input as RSPCK (the maximum frequency of RSPCK is that of PCLK divided by 8).<br>Width at high level: 4 cycles of PCLK; width at low level: 4 cycles of PCLK |
| Buffer configuration | • Double buffer configuration for the transmit/receive buffers<br>• 128 bits for the transmit/receive buffers |
| Error detection | • Mode fault error detection<br>• Overrun error detection<br>• Parity error detection |
| SSL control function | • Four SSL pins (SSLA0 to SSLA3) for each channel<br>• In single-master mode, SSLA0 to SSLA3 pins are output.<br>• In multi-master mode:<br>SSLA0 pin for input, and SSLA1 to SSLA3 pins for either output or unused.<br>• In slave mode:<br>SSLA0 pin for input, and SSLA1 to SSLA3 pins for unused.<br>• Controllable delay from SSL output assertion to RSPCK operation (RSPCK delay)<br>Range: 1 to 8 RSPCK cycles (set in RSPCK-cycle units)<br>• Controllable delay from RSPCK stop to SSL output negation (SSL negation delay)<br>Range: 1 to 8 RSPCK cycles (set in RSPCK-cycle units)<br>• Controllable wait for next-access SSL output assertion (next-access delay)<br>Range: 1 to 8 RSPCK cycles (set in RSPCK-cycle units)<br>• Function for changing SSL polarity |
| Control in master transfer | • A transfer of up to eight commands can be executed sequentially in looped execution.<br>• For each command, the following can be set:<br>SSL signal value, bit rate, RSPCK polarity/phase, transfer data length, MSB/LSB first, burst, RSPCK delay, SSL negation delay, and next-access delay<br>• A transfer can be initiated by writing to the transmit buffer.<br>• MOSI signal value specifiable in SSL negation |
| Interrupt sources | • Interrupt sources<br>Receive buffer full interrupt<br>Transmit buffer empty interrupt<br>RSPI error interrupt (mode fault, overrun, parity error)<br>RSPI idle interrupt (RSPI idle) |

**Table 28.1 RSPI Specifications (2/2)**

| Item | Description |
|---|---|
| Others | • Function for switching between CMOS output and open-drain output<br>• Function for initializing the RSPI<br>• Loopback mode |
| Low power consumption function | Module stop state can be set. |




SPCR: RSPI control register
SPCR2: RSPI control register 2
SSLP: RSPI slave select polarity register
SPPCR: RSPI pin control register
SPSR: RSPI status register
SPDR: RSPI data register
SPSCR: RSPI sequence control register
SPSSR: RSPI sequence status register
SPDCR: RSPI data control register
SPCKD: RSPI clock delay register
SSLND: RSPI slave select negation delay register
SPND: RSPI next-access delay register
SPCMD: RSPI command registers 0 to 7 (eight registers)
SPBR: RSPI bit rate register
SPTX: RSPI transmit buffer
SPRX: RSPI receive buffer

**Figure 28.1 RSPI Block Diagram**

Table 28.2 lists the I/O pins used in the RSPI.
The RSPI automatically switches the I/O direction of the SSLA0 pin. SSLA0 is set as an output when the RSPI is a single master and as an input when the RSPI is a multi-master or a slave. Pins RSPCKA, MOSIA, and MISOA are automatically set as inputs or outputs according to the setting of master or slave and the level input on the SSLA0 pin (refer to section 28.3.2, Controlling RSPI Pins).

**Table 28.2 RSPI Pin Configuration**

| Channel | Pin Name | I/O | Function |
|---|---|---|---|
| RSPI0 | RSPCKA | I/O | Clock I/O |
| | MOSIA | I/O | Master transmit data I/O |
| | MISOA | I/O | Slave transmit data I/O |
| | SSLA0 | I/O | Slave selection I/O |
| | SSLA1 | Output | Slave selection output |
| | SSLA2 | Output | Slave selection output |
| | SSLA3 | Output | Slave selection output |

## 28.2 Register Descriptions

### 28.2.1 RSPI Control Register (SPCR)

Address(es): RSPI0.SPCR 0008 8380h

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | SPRIE | SPE | SPTIE | SPEIE | MSTR | MODFEN | TXMD | SPMS |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b0 | SPMS | RSPI Mode Select | 0: SPI operation (four-wire method)<br>1: Clock synchronous operation (three-wire method) | R/W |
| b1 | TXMD | Communications Operating Mode Select | 0: Full-duplex synchronous serial communications<br>1: Serial communications consisting of only transmit operations | R/W |
| b2 | MODFEN | Mode Fault Error Detection Enable | 0: Disables the detection of mode fault error<br>1: Enables the detection of mode fault error | R/W |
| b3 | MSTR | RSPI Master/Slave Mode Select | 0: Slave mode<br>1: Master mode | R/W |
| b4 | SPEIE | RSPI Error Interrupt Enable | 0: Disables the generation of RSPI error interrupt requests<br>1: Enables the generation of RSPI error interrupt requests | R/W |
| b5 | SPTIE | Transmit Buffer Empty Interrupt Enable | 0: Disables the generation of transmit buffer empty interrupt requests<br>1: Enables the generation of transmit buffer empty interrupt requests | R/W |
| b6 | SPE | RSPI Function Enable | 0: Disables the RSPI function<br>1: Enables the RSPI function | R/W |
| b7 | SPRIE | RSPI Receive Interrupt Enable | 0: Disables the generation of RSPI receive interrupt requests<br>1: Enables the generation of RSPI receive interrupt requests | R/W |

If the SPCR.MSTR, SPCR.MODFEN, or SPCR.TXMD bit is changed while the SPCR.SPE bit is 1, subsequent operations should not be performed.

**SPMS Bit (RSPI Mode Select)**

The SPMS bit selects SPI operation (four-wire method) or clock synchronous operation (three-wire method).
The SSLA0 to SSLA3 pins are not used in clock synchronous operation. The three pins RSPCKA, MOSIA, and MISOA handle communications. If clock synchronous operation is to proceed in master mode (SPCR.MSTR = 1), the SPCMDm.CPHA bit can be set to either 0 or 1. Set the CPHA bit to 1 if clock synchronous operation is to proceed in slave mode (SPCR.MSTR = 0). Operation should not be performed if the CPHA bit is set to 0 when clock synchronous operation is to proceed in slave mode (SPCR.MSTR = 0).

**TXMD Bit (Communications Operating Mode Select)**

The TXMD bit selects full-duplex synchronous serial communications or transmit operations only.
When performing communications with the TXMD bit set to 1, the RSPI performs only transmit operations and not receive operations (refer to section 28.3.6, Communications Operating Mode).
When the TXMD bit is set to 1, receive buffer full interrupt requests cannot be used.

#### MODFEN Bit (Mode Fault Error Detection Enable)

The MODFEN bit enables or disables the detection of mode fault error (refer to section 28.3.8, Error Detection). In addition, the RSPI determines the I/O direction of the SSLA0 to SSLA3 pins based on combinations of the MODFEN and MSTR bits (refer to section 28.3.2, Controlling RSPI Pins).

#### MSTR Bit (RSPI Master/Slave Mode Select)

The MSTR bit selects master/slave mode of the RSPI. According to MSTR bit settings, the RSPI determines the direction of pins RSPCKA, MOSIA, MISOA, and SSLA0 to SSLA3.

#### SPEIE Bit (RSPI Error Interrupt Enable)

The SPEIE bit enables or disables the generation of RSPI error interrupt requests when the RSPI detects a mode fault error and sets the SPSR.MODF flag to 1, when the RSPI detects an overrun error and sets the SPSR.OVRF flag to 1, or when the RSPI detects a parity error and sets the SPSR.PERF flag to 1 (refer to section 28.3.8, Error Detection).

#### SPTIE Bit (Transmit Buffer Empty Interrupt Enable)

The SPTIE bit enables or disables the generation of transmit buffer empty interrupt requests when the RSPI detects when the transmit buffer is empty.
A transmit buffer empty interrupt request when transmission starts is generated by setting the SPE and SPTIE bits to 1 at the same time or by setting the SPE bit to 1 after setting the SPTIE bit to 1.
Note that a transmit buffer interrupt is generated when the SPTIE bit is 1 even if the RSRI function is disabled (the SPTIE bit is changed to 0).

#### SPE Bit (RSPI Function Enable)

The SPE bit enables or disables the RSPI function.
When the SPSR.MODF bit is 1, the SPE bit cannot be set to 1. For details, refer to section 28.3.8, Error Detection.
Setting the SPE bit to 0 disables the RSPI function, and initializes a part of the module function. For details, refer to section 28.3.9, Initializing RSPI. Furthermore, a transmit buffer empty interrupt request is generated by the state of the SPE bit changing from 0 to 1 or from 1 to 0.

#### SPRIE Bit (RSPI Receive Interrupt Enable)

The SPRIE bit enables or disables the generation of RSPI receive interrupt requests when the RSPI detects when data is written to the receive buffer after completion of serial transfer.

## 28.2.2 RSPI Slave Select Polarity Register (SSLP)

Address(es): RSPI0.SSLP 0008 8381h

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | SSL3P | SSL2P | SSL1P | SSL0P |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b0 | SSL0P | SSL0 Signal Polarity Setting | 0: SSL0 signal is active low<br>1: SSL0 signal is active high | R/W |
| b1 | SSL1P | SSL1 Signal Polarity Setting | 0: SSL1 signal is active low<br>1: SSL1 signal is active high | R/W |
| b2 | SSL2P | SSL2 Signal Polarity Setting | 0: SSL2 signal is active low<br>1: SSL2 signal is active high | R/W |
| b3 | SSL3P | SSL3 Signal Polarity Setting | 0: SSL3 signal is active low<br>1: SSL3 signal is active high | R/W |
| b7 to b4 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |

If the contents of SSLP are changed while the SPCR.SPE bit is 1, subsequent operations should not be performed.

## 28.2.3 RSPI Pin Control Register (SPPCR)

Address(es): RSPI0.SPPCR 0008 8382h

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | MOIFE | MOIFV | — | — | SPLP2 | SPLP |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b0 | SPLP | RSPI Loopback | 0: Normal mode<br>1: Loopback mode (data is inverted for transmission) | R/W |
| b1 | SPLP2 | RSPI Loopback 2 | 0: Normal mode<br>1: Loopback mode (data is not inverted for transmission) | R/W |
| b3, b2 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |
| b4 | MOIFV | MOSI Idle Fixed Value | 0: The level output on the MOSIA pin during MOSI idling corresponds to low.<br>1: The level output on the MOSIA pin during MOSI idling corresponds to high. | R/W |
| b5 | MOIFE | MOSI Idle Value Fixing Enable | 0: MOSI output value equals final data from previous transfer<br>1: MOSI output value equals the value set in the MOIFV bit | R/W |
| b7, b6 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |

If the contents of SPPCR are changed while the SPCR.SPE bit is 1, subsequent operations should not be performed.

**SPLP Bit (RSPI Loopback)**

The SPLP bit selects the mode of the RSPI pins.

When the SPLP bit is set to 1, the RSPI shuts off the path between the MISOA pin and the shift register if the SPCR.MSTR bit is 1, and between the MOSIA pin and the shift register if the SPCR.MSTR bit is 0, and connects (reverses) the input path and output path for the shift register (loopback mode).

**SPLP2 Bit (RSPI Loopback 2)**

The SPLP2 bit selects the mode of the RSPI pins.

When the SPLP2 bit is set to 1, the RSPI shuts off the path between the MISOA pin and the shift register if the SPCR.MSTR bit is 1, and between the MOSIA pin and the shift register if the SPCR.MSTR bit is 0, and connects the input path and output path for the shift register (loopback mode).

**MOIFV Bit (MOSI Idle Fixed Value)**

If the MOIFE bit is 1 in master mode, the MOIFV bit determines the MOSIA pin output value during the SSL negation period (including the SSL retention period during a burst transfer).

**MOIFE Bit (MOSI Idle Value Fixing Enable)**

The MOIFE bit fixes the MOSIA output value when the RSPI in master mode is in an SSL negation period (including the SSL retention period during a burst transfer). When the MOIFE bit is 0, the RSPI outputs the last data from the previous serial transfer during the SSL negation period to the MOSIA pin. When the MOIFE bit is 1, the RSPI outputs the fixed value set in the MOIFV bit to the MOSIA pin.

## 28.2.4 RSPI Status Register (SPSR)

Address(es): RSPI0.SPSR 0008 8383h

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | PERF | MODF | IDLNF | OVRF |
| Value after reset: | x | 0 | x | 0 | 0 | 0 | 0 | 0 |

x: Undefined

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b0 | OVRF | Overrun Error Flag | 0: No overrun error occurs<br>1: An overrun error occurs | R/(W) *1 |
| b1 | IDLNF | RSPI Idle Flag | 0: RSPI is in the idle state<br>1: RSPI is in the transfer state | R |
| b2 | MODF | Mode Fault Error Flag | 0: No mode fault error occurs<br>1: A mode fault error occurs | R/(W) *1 |
| b3 | PERF | Parity Error Flag | 0: No parity error occurs<br>1: A parity error occurs | R/(W) *1 |
| b4 | — | Reserved | This bit is read as 0. The write value should be 0. | R/W |
| b5 | — | Reserved | The read value is undefined. The write value should be 1. | R/W |
| b6 | — | Reserved | This bit is read as 0. The write value should be 0. | R/W |
| b7 | — | Reserved | The read value is undefined. The write value should be 1. | R/W |

Note 1. Only 0 can be written to clear the flag after reading 1.

### OVRF Flag (Overrun Error Flag)

The OVRF flag indicates the occurrence of an overrun error.

[Setting condition]

- When the next serial transfer ends while the SPCR.TXMD bit is 0 and the receive buffer is full.

[Clearing condition]

- When SPSR is read while the OVRF flag is 1, and then writes the value 0 to the OVRF flag.

### IDLNF Flag (RSPI Idle Flag)

The IDLNF flag indicates the transfer status of the RSPI.

[Setting condition]

Master mode

- Condition 1 and condition 2 are not satisfied in master mode under the [Clearing condition] below.

Slave mode

- The SPCR.SPE bit is 1 (RSPI function is enabled)

[Clearing condition]

Master mode

- The following 1 is satisfied (condition 1) or all of the following 2 to 4 are satisfied (condition 2).

1. The SPCR.SPE bit is 0 (RSPI is initialized)
2. The transmit buffer (SPTX) is empty (data for the next transfer is not set)
3. The SPSSR.SPCP[2:0] bits are 000b (beginning of sequence control)
4. The RSPI internal sequencer has entered the idle state (status in which operations up to the next-access delay have finished)

Slave mode

- The SPCR.SPE bit is 0 (RSPI is initialized)

### MODF Flag (Mode Fault Error Flag)

Indicates the occurrence of a mode fault error.

[Setting condition]

Multi-master mode

- When the input level of the SSLAi pin changes to the active level while the SPCR.MSTR bit is 1 (master mode) and the SPCR.MODFEN bit is 1 (mode fault error detection is enabled), the RSPI detects a mode fault error

Slave mode

- When the SSLAi pin is negated before the RSPCK cycle necessary for data transfer ends while the SPCR.MSTR bit is 0 (slave mode) and the SPCR.MODFEN bit is 1 (mode fault error detection is enabled), the RSPI detects a mode fault error

The active level of the SSLAi signal is determined by the SSLP.SSLiP bit (SSLi signal polarity setting bit).

[Clearing condition]

- When SPSR is read while the MODF flag is 1, and then writes the value 0 to the MODF flag

### PERF Flag (Parity Error Flag)

Indicates the occurrence of a parity error.

[Setting condition]

- When a serial transfer ends while the SPCR.TXMD bit is 0 and the SPCR2.SPPE bit is 1, the RSPI detects a parity error

[Clearing condition]

- When SPSR is read while the PERF flag is 1, and then writes the value 0 to the PERF flag

## 28.2.5 RSPI Data Register (SPDR)

Address(es): RSPI0.SPDR 0008 8384h

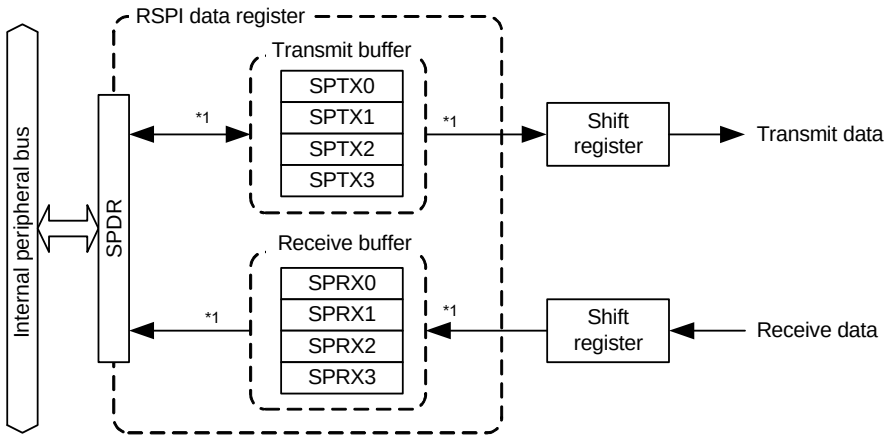
| | b31 | b30 | b29 | b28 | b27 | b26 | b25 | b24 | b23 | b22 | b21 | b20 | b19 | b18 | b17 | b16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Address(es): RSPI0.SPDR 0008 8384h

| | b31 | b30 | b29 | b28 | b27 | b26 | b25 | b24 | b23 | b22 | b21 | b20 | b19 | b18 | b17 | b16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | | | H | | | | | | | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

SPDR is the interface with the buffers that hold data for transmission and reception by the RSPI.
When accessing in longwords (the SPLW bit is 1), access SPDR.
When accessing in words (the SPLW bit is 0), access the higher-order 16 bits (H) of SPDR.
The transmit buffer (SPTX) and receive buffer (SPRX) are independent but are both mapped to SPDR. Figure 28.2 shows the Configuration of SPDR.




Note 1. The destination buffer and stage for access is automatically switched by hardware.

**Figure 28.2 Configuration of SPDR**

The transmit and receive buffers each have four stages. The number of stages to be used is selectable by the number of frames specification bits in the RSPI data control register (SPDCR.SPFC[1:0]). The eight stages of the buffer are all mapped to the single address of SPDR.
Data written to SPDR are written to a transmit-buffer stage (SPTXn) (n = 0 to 3) and then transmitted from the buffer.
The receive buffer holds received data on completion of reception. The receive buffer is not updated if an overrun is generated.

Furthermore, if the data length is other than 32 bits, bits not referred to in SPTXn (n = 0 to 3) are stored in the corresponding bits in SPRXn. For example, if the data length is 9 bits, received data are stored in the SPRXn[8:0] bits and the SPTXn[31:9] bits are stored in the SPRXn[31:9] bits.

### (1) Bus Interface

SPDR is the interface with 32-bit wide transmit and receive buffers, each of which has four stages, for a total of 32 bytes. In other words, the 32 bytes are mapped to the 4-byte address space for SPDR. Furthermore, the unit of access for SPDR is selected by the RSPI longword access/word access specification bit in the RSPI data control register (SPDCR.SPLW). Data for transmission should be flush with the LSB end of the register. Received data are stored flush with the LSB end. Operations involved in writing to and reading from SPDR are described below.

#### (a) Writing

Data written to SPDR are written to a transmit buffer (SPTXn). This is not influenced by the value of the SPDCR.SPRDTD bit unlike when reading from SPDR.
The transmit buffer includes a transmit buffer write pointer which is automatically updated to indicate the next stage each time data are written to SPDR.
Figure 28.3 shows the configuration of the bus interface with the transmit buffer in the case of writing to SPDR.

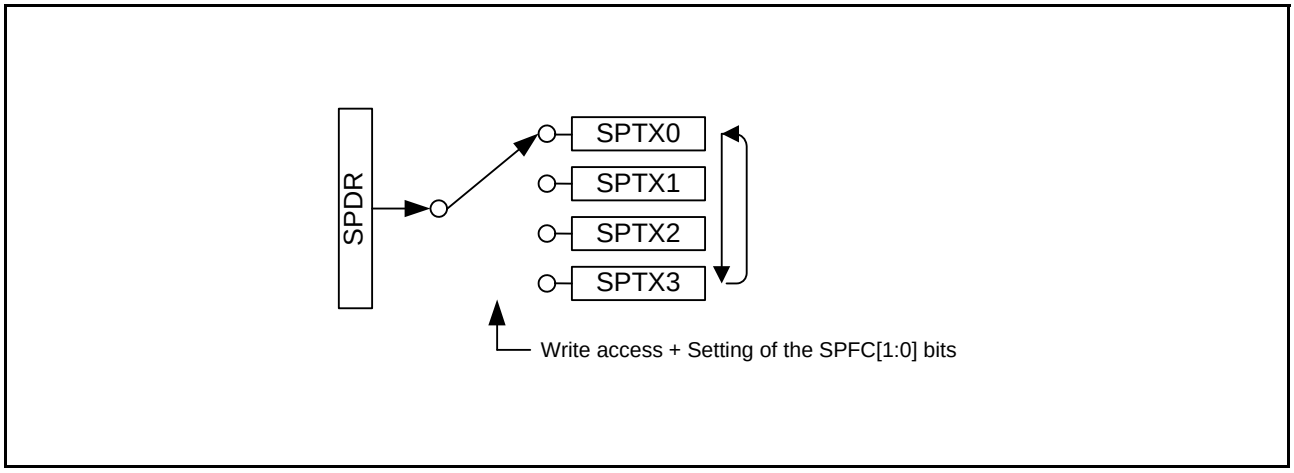



**Figure 28.3 Configuration of SPDR (Writing)**

The sequence for switching the transmit buffer write pointer differs with the setting of the number of frames specification bits in the RSPI data control register (SPDCR.SPFC[1:0]).

- Settings of the SPFC[1:0] bits and sequence of switching the pointer among SPTX0 to SPTX3.
  When the SPFC[1:0] bits are 00b: SPTX0 → SPTX0 → SPTX0 → …
  When the SPFC[1:0] bits are 01b: SPTX0 → SPTX1 → SPTX0 → SPTX1 → …
  When the SPFC[1:0] bits are 10b: SPTX0 → SPTX1 → SPTX2 → SPTX0 → SPTX1 → …
  When the SPFC[1:0] bits are 11b: SPTX0 → SPTX1 → SPTX2 → SPTX3 → SPTX0 → SPTX1 → …

When 1 is written to the RSPI function enable bit in the RSPI control register (SPCR.SPE) while the bit's current value is 0, SPTX0 will be the destination the next time writing proceeds.
When writing to the transmission buffer (SPTXn) after generation of the transmit buffer empty interrupt, write the number of frames set by the number of frames specification bits (SPFC[1:0]) in the RSPI data control register (SPDCR). Even if the number of frames is written to the transmit buffer (SPTXn), the value of the buffer is not updated after completion of the writing and before generation of the next transmit buffer empty interrupt.

### (b) Reading

SPDR can be read to read the value of a receive buffer (SPRXn) or a transmit buffer (SPTXn). The setting of the RSPI receive/transmit data selection bit in the RSPI data control register (SPDCR.SPRDTD) selects whether reading is of the receive or transmit buffer.

The sequence of reading the SPDR register is controlled by independent pointers, receive buffer read pointer and transmit buffer read pointer.

Figure 28.4 shows the configuration of the bus interface with the receive and transmit buffers in the case of reading from SPDR.

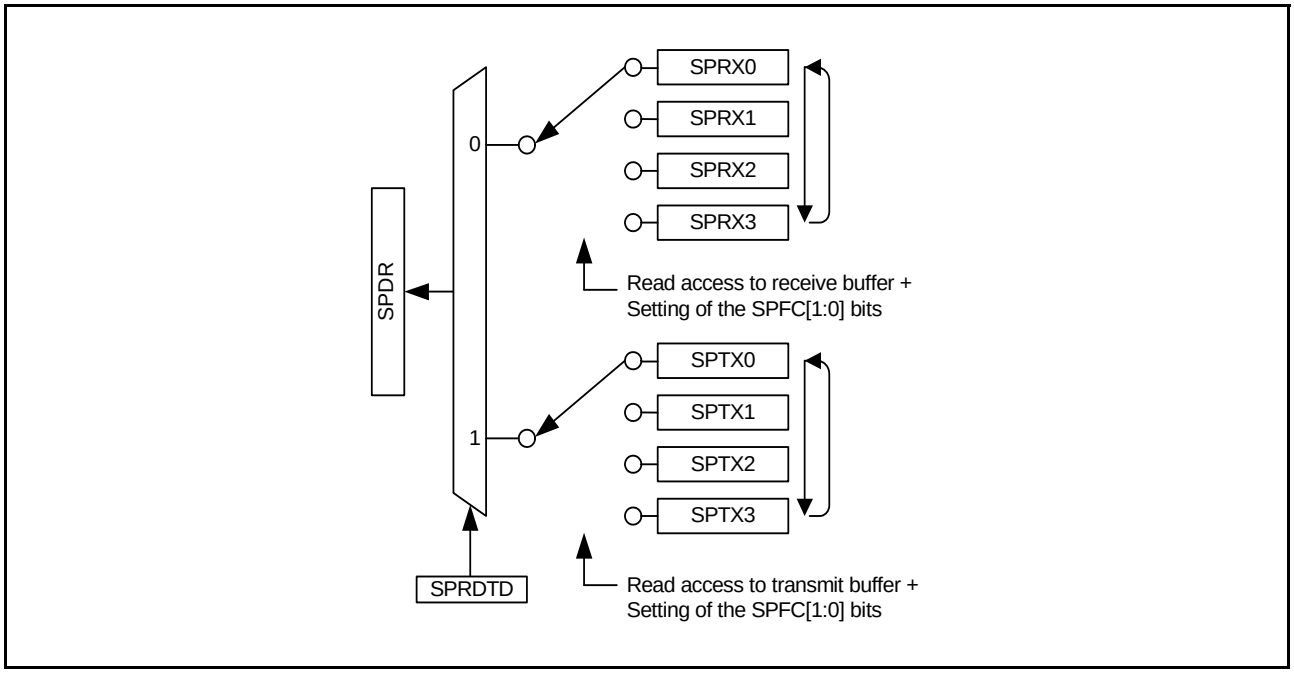



**Figure 28.4 Configuration of SPDR (Reading)**

Reading the receive buffer switches the receive buffer read pointer to the next buffer automatically.

The sequence of switching the receive buffer read pointer is the same as that for the transmit buffer write pointer. However, when 1 is written to the RSRI function enable bit in the RSPI control register (SPCR.SPE) while the bit's current value is 1, SPRX0 will be indicated by the buffer read pointer the next time reading proceeds.

The transmit buffer read pointer is updated when writing to SPDR, and not updated when reading from the transmit buffer. When reading from the transmit buffer, the value most recently written to SPDR is read. However, after generation of the transmit buffer empty interrupt, the values read from the transmit buffer are all 0 in the interval after completion of writing the number of frames of data specified in the number of frames specification bits (SPDCR.SPFC[1:0]) and before generation of the next buffer empty interrupt.

## 28.2.6 RSPI Sequence Control Register (SPSCR)

Address(es): RSPI0.SPSCR 0008 8388h

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | SPSLN[2:0] | | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b2 to b0 | SPSLN[2:0] | RSPI Sequence Length Specification | b2 b0 Sequence Length — Referenced SPCMD0 to SPCMD7 (No.)<br>0 0 0: 1 — 0→0→…<br>0 0 1: 2 — 0→1→0→…<br>0 1 0: 3 — 0→1→2→0→…<br>0 1 1: 4 — 0→1→2→3→0→…<br>1 0 0: 5 — 0→1→2→3→4→0→…<br>1 0 1: 6 — 0→1→2→3→4→5→0→…<br>1 1 0: 7 — 0→1→2→3→4→5→6→0→…<br>1 1 1: 8 — 0→1→2→3→4→5→6→7→0→…<br>The order in which the SPCMD0 to SPCMD07 registers are to be referenced is changed in accordance with the sequence length that is set in these bits. The relationship among the setting of these bits, sequence length, and SPCMD0 to SPCMD7 registers referenced by the RSPI is shown above. However, the RSPI in slave mode always references SPCMD0. | R/W |
| b7 to b3 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |

SPSCR sets the sequence length when the RSPI operates in master mode. When changing the SPSCR.SPSLN[2:0] bits while both the SPCR.MSTR and SPCR.SPE bits are 1, the bits should be changed while the SPSR.IDLNF flag is 0.

### SPSLN[2:0] Bits (RSPI Sequence Length Specification)

The SPSLN[2:0] bits specify a sequence length when the RSPI in master mode performs sequential operations. The RSPI in master mode changes SPCMD0 to SPCMD7 registers to be referenced and the order in which they are referenced according to the sequence length that is set in the SPSLN[2:0] bits.
In slave mode, SPCMD0 is always referred to.

## 28.2.7 RSPI Sequence Status Register (SPSSR)

Address(es): RSPI0.SPSSR 0008 8389h

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | SPECM[2:0] | | | — | SPCP[2:0] | | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b2 to b0 | SPCP[2:0] | RSPI Command Pointer | b2 b0<br>0 0 0: SPCMD0<br>0 0 1: SPCMD1<br>0 1 0: SPCMD2<br>0 1 1: SPCMD3<br>1 0 0: SPCMD4<br>1 0 1: SPCMD5<br>1 1 0: SPCMD6<br>1 1 1: SPCMD7 | R |
| b3 | — | Reserved | This bit is read as 0. | R |
| b6 to b4 | SPECM[2:0] | RSPI Error Command | b6 b4<br>0 0 0: SPCMD0<br>0 0 1: SPCMD1<br>0 1 0: SPCMD2<br>0 1 1: SPCMD3<br>1 0 0: SPCMD4<br>1 0 1: SPCMD5<br>1 1 0: SPCMD6<br>1 1 1: SPCMD7 | R |
| b7 | — | Reserved | This bit is read as 0. | R |

SPSSR indicates the sequence control status when the RSPI operates in master mode.
Any writing to SPSSR is ignored.

### SPCP[2:0] Bits (RSPI Command Pointer)

The SPCP[2:0] bits indicate SPCMDm that is currently pointed to by the pointer during sequence control by the RSPI. For the RSPI's sequence control, refer to section 28.3.10.1, Master Mode Operation.

### SPECM[2:0] Bits (RSPI Error Command)

The SPECM[2:0] bits indicate SPCMDm that is specified by the SPCP[2:0] bits when an error is detected during sequence control by the RSPI. The RSPI updates the SPECM[2:0] bits only when an error is detected. If both the SPSR.OVRF and SPSR.MODF bits are 0 and there is no error, the values of the SPECM[2:0] bits have no meaning. For the RSPI's error detection function, refer to section 28.3.8, Error Detection. For the RSPI's sequence control, refer to section 28.3.10.1, Master Mode Operation.

## 28.2.8 RSPI Bit Rate Register (SPBR)

Address(es): RSPI0.SPBR 0008 838Ah

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| Value after reset: | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

SPBR sets the bit rate in master mode. If the contents of SPBR are changed while both the SPCR.MSTR and SPCR.SPE bits are 1, subsequent operations should not be performed.

When the RSPI is used in slave mode, the bit rate depends on the bit rate of the input clock (bit rate satisfying the electrical characteristics should be used) regardless of the settings of SPBR and the SPCMDm.BRDV[1:0] bits (bit rate division setting bits).
The bit rate is determined by combinations of the SPBR setting and the SPCMDm.BRDV[1:0] bit setting. The equation for calculating the bit rate is given below. In the equation, n denotes an SPBR setting (0, 1, 2, …, 255), and N denotes a BRDV[1:0] bit setting (0, 1, 2, 3).

$$\text{Bit rate} = \frac{f(\text{PCLK})}{2 \times (n + 1)\, 2^{N}}$$

Table 28.3 lists examples of the relationship among the SPBR settings, the BRDV[1:0] settings, and bit rates.

**Table 28.3 Relationship among SPBR Settings, BRDV[1:0] Settings, and Bit Rates**

| SPBR (n) | BRDV[1:0] Bits (N) | Division Ratio | Bit Rate PCLK = 32 MHz |
|---|---|---|---|
| 0 | 0 | 2 | 16.0 Mbps |
| 1 | 0 | 4 | 8.00 Mbps |
| 2 | 0 | 6 | 5.33 Mbps |
| 3 | 0 | 8 | 4.00 Mbps |
| 4 | 0 | 10 | 3.20 Mbps |
| 5 | 0 | 12 | 2.67 Mbps |
| 5 | 1 | 24 | 1.33 Mbps |
| 5 | 2 | 48 | 667 kbps |
| 5 | 3 | 96 | 333 kbps |
| 255 | 3 | 4096 | 7.81 kbps |

## 28.2.9 RSPI Data Control Register (SPDCR)

Address(es): RSPI0.SPDCR 0008 838Bh

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | SPLW | SPRDTD | — | — | SPFC[1:0] | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b1, b0 | SPFC[1:0] | Number of Frames Specification | b1 b0<br>0 0: 1 frame<br>0 1: 2 frames<br>1 0: 3 frames<br>1 1: 4 frames | R/W |
| b3, b2 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |
| b4 | SPRDTD | RSPI Receive/Transmit Data Selection | 0: SPDR values are read from the receive buffer<br>1: SPDR values are read from the transmit buffer (but only if the transmit buffer is empty) | R/W |
| b5 | SPLW | RSPI Longword Access/ Word Access Specification | 0: SPDR is accessed in words<br>1: SPDR is accessed in longwords | R/W |
| b7, b6 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |

Up to four frames can be transmitted or received in one round of transmission or reception activation. The amount of data in each transfer is controlled by the combination of the SPCMDm.SPB[3:0] bits, the SPSCR.SPSLN[2:0] bits, and the SPDCR.SPFC[1:0] bits.
When changing the SPDCR.SPFC[1:0] bits while the SPCR.SPE bit is 1, the bits should be changed while the SPSR.IDLNF flag is 0.

### SPFC[1:0] Bits (Number of Frames Specification)

The SPFC[1:0] bits specify the number of frames that can be stored in SPDR (per transfer activation). Up to four frames can be transmitted or received in one round of transmission or reception, and the amount of data is determined by the combination of the SPSCR.SPSLN[2:0] bits, and the SPDCR.SPFC[1:0] bits. Furthermore, the setting of the SPFC[1:0] bits adjusts the number of frames for generation of RSPI reception interrupts, and start of transmission or generation of transmit buffer empty interrupts. Table 28.4 lists the frame configurations that can be stored in SPDR and examples of combinations of settings for transmission and reception. If combinations of settings other than those shown in the examples are made, subsequent operations should not be performed.

**Table 28.4 Settable Combinations of SPSLN[2:0] Bits and SPFC[1:0] Bits**

| Setting | SPSLN[2:0] | SPFC[1:0] | Number of Frames in a Single Sequence | Number of Frames at which Receive Buffer Full Interrupt Occurs or Transmit Buffer Holding Data is Recognized |
|---|---|---|---|---|
| 1-1 | 000b | 00b | 1 | 1 |
| 1-2 | 000b | 01b | 2 | 2 |
| 1-3 | 000b | 10b | 3 | 3 |
| 1-4 | 000b | 11b | 4 | 4 |
| 2-1 | 001b | 01b | 2 | 2 |
| 2-2 | 001b | 11b | 4 | 4 |
| 3 | 010b | 10b | 3 | 3 |
| 4 | 011b | 11b | 4 | 4 |
| 5 | 100b | 00b | 5 | 1 |
| 6 | 101b | 00b | 6 | 1 |
| 7 | 110b | 00b | 7 | 1 |
| 8 | 111b | 00b | 8 | 1 |

### SPRDTD Bit (RSPI Receive/Transmit Data Selection)

The SPRDTD bit selects whether the SPDR reads values from the receive buffer or from the transmit buffer.
If reading is from the transmit buffer, the value written to SPDR register immediately beforehand is read.
When reading the transmit buffer, do so before writing of the number of frames set in the SPFC[1:0] bits is finished and after generation of the transmit buffer empty interrupt.
For details, refer to section 28.2.5, RSPI Data Register (SPDR).

### SPLW Bit (RSPI Longword Access/Word Access Specification)

The SPLW bit specifies the access width for SPDR. Access to SPDR is in words when the SPLW bit is 0 and in longwords when the SPLW bit is 1.
Also, when the SPLW bit is 0, set the SPCMDm.SPB[3:0] bits (RSPI data length setting bits) to 8 to 16 bits. When 20, 24, or 32 bits is specified, operations should not be performed.

## 28.2.10 RSPI Clock Delay Register (SPCKD)

Address(es): RSPI0.SPCKD 0008 838Ch

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | SCKDL[2:0] | | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b2 to b0 | SCKDL[2:0] | RSPCK Delay Setting | b2 b1 b0<br>0 0 0: 1 RSPCK<br>0 0 1: 2 RSPCK<br>0 1 0: 3 RSPCK<br>0 1 1: 4 RSPCK<br>1 0 0: 5 RSPCK<br>1 0 1: 6 RSPCK<br>1 1 0: 7 RSPCK<br>1 1 1: 8 RSPCK | R/W |
| b7 to b3 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |

SPCKD sets a period from the beginning of SSLAi signal assertion to RSPCK oscillation (RSPCK delay) when the SPCMDm.SCKDEN bit is 1. If the contents of SPCKD are changed while both the SPCR.MSTR and SPCR.SPE bits are 1, subsequent operations should not be performed.

**SCKDL[2:0] Bits (RSPCK Delay Setting)**

The SCKDL[2:0] bits set an RSPCK delay value when the SPCMDm.SCKDEN bit is 1.
When using the RSPI in slave mode, set the SCKDL[2:0] bits to 000b.

## 28.2.11 RSPI Slave Select Negation Delay Register (SSLND)

Address(es): RSPI0.SSLND 0008 838Dh

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | SLNDL[2:0] | | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b2 to b0 | SLNDL[2:0] | SSL Negation Delay Setting | b2 b1 b0<br>0 0 0: 1 RSPCK<br>0 0 1: 2 RSPCK<br>0 1 0: 3 RSPCK<br>0 1 1: 4 RSPCK<br>1 0 0: 5 RSPCK<br>1 0 1: 6 RSPCK<br>1 1 0: 7 RSPCK<br>1 1 1: 8 RSPCK | R/W |
| b7 to b3 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |

SSLND sets a period (SSL negation delay) from the transmission of a final RSPCK edge to the negation of the SSLAi signal during a serial transfer by the RSPI in master mode. If the contents of SSLND are changed while both the SPCR.MSTR and SPCR.SPE bits are 1, subsequent operations should not be performed.

**SLNDL[2:0] Bits (SSL Negation Delay Setting)**

The SLNDL[2:0] bits set an SSL negation delay value when the RSPI is in master mode.
When using the RSPI in slave mode, set the SLNDL[2:0] bits to 000b.

## 28.2.12 RSPI Next-Access Delay Register (SPND)

Address(es): RSPI0.SPND 0008 838Eh

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | SPNDL[2:0] | | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b2 to b0 | SPNDL[2:0] | RSPI Next-Access Delay Setting | b2 b0<br>0 0 0: 1 RSPCK + 2 PCLK<br>0 0 1: 2 RSPCK + 2 PCLK<br>0 1 0: 3 RSPCK + 2 PCLK<br>0 1 1: 4 RSPCK + 2 PCLK<br>1 0 0: 5 RSPCK + 2 PCLK<br>1 0 1: 6 RSPCK + 2 PCLK<br>1 1 0: 7 RSPCK + 2 PCLK<br>1 1 1: 8 RSPCK + 2 PCLK | R/W |
| b7 to b3 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |

SPND sets a non-active period (next-access delay) of the SSLAi signal after termination of a serial transfer when the SPCMDm.SPNDEN bit is 1. If the contents of SPND are changed while both the SPCR.MSTR and SPCR.SPE bits are 1, subsequent operations should not be performed.

**SPNDL[2:0] Bits (RSPI Next-Access Delay Setting)**

The SPNDL[2:0] bits set a next-access delay when the SPCMDm.SPNDEN bit is 1.
When using the RSPI in slave mode, set the SPNDL[2:0] bits to 000b.

## 28.2.13 RSPI Control Register 2 (SPCR2)

Address(es): RSPI0.SPCR2 0008 838Fh

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | PTE | SPIIE | SPOE | SPPE |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b0 | SPPE | Parity Enable | 0: Does not add the parity bit to transmit data and does not check the parity bit of receive data<br>1: Adds the parity bit to transmit data and checks the parity bit of receive data (when SPCR.TXMD = 0)<br>Adds the parity bit to transmit data but does not check the parity bit of receive data (when SPCR.TXMD = 1) | R/W |
| b1 | SPOE | Parity Mode | 0: Selects even parity for use in transmission and reception<br>1: Selects odd parity for use in transmission and reception | R/W |
| b2 | SPIIE | RSPI Idle Interrupt Enable | 0: Disables the generation of idle interrupt requests<br>1: Enables the generation of idle interrupt requests | R/W |
| b3 | PTE | Parity Self-Testing | 0: Disables the self-diagnosis function of the parity circuit<br>1: Enables the self-diagnosis function of the parity circuit | R/W |
| b7 to b4 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |

If the SPPE bit or SPOE bit in SPCR2 is changed while the SPCR.SPE bit is 1, subsequent operations should not be performed.

### SPPE Bit (Parity Enable)

The SPPE bit enables or disables the parity function.

The parity bit is added to transmit data and parity checking is performed for receive data when the SPCR.TXMD bit is 0 and the SPCR2.SPPE bit is 1.

The parity bit is added to transmit data but parity checking is not performed for receive data when the SPCR.TXMD bit is 1 and the SPCR2.SPPE bit is 1.

### SPOE Bit (Parity Mode)

The SPOE bit specifies odd or even parity.

When even parity is set, parity bit addition is performed so that the total number of 1-bits in the transmit/receive character plus the parity bit is even. Similarly, when odd parity is set, parity bit addition is performed so that the total number of 1-bits in the transmit/receive character plus the parity bit is odd.

The SPOE bit is valid only when the SPPE bit is 1.

### SPIIE Bit (RSPI Idle Interrupt Enable)

The SPIIE bit enables or disables the generation of RSPI idle interrupt requests when the RSPI being in the idle state is detected and the SPSR.IDLNF flag is cleared to 0.

### PTE Bit (Parity Self-Testing)

The PTE bit enables the self-diagnosis function of the parity circuit in order to check whether the parity function is operating correctly.

## 28.2.14 RSPI Command Registers 0 to 7 (SPCMD0 to SPCMD7)

Address(es): RSPI0.SPCMD0 0008 8390h, RSPI0.SPCMD1 0008 8392h, RSPI0.SPCMD2 0008 8394h, RSPI0.SPCMD3 0008 8396h, RSPI0.SPCMD4 0008 8398h, RSPI0.SPCMD5 0008 839Ah, RSPI0.SPCMD6 0008 839Ch, RSPI0.SPCMD7 0008 839Eh

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | SCKDEN | SLNDEN | SPNDEN | LSBF | SPB[3:0] | | | | SSLKP | SSLA[2:0] | | | BRDV[1:0] | | CPOL | CPHA |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b0 | CPHA | RSPCK Phase Setting | 0: Data sampling on odd edge, data variation on even edge<br>1: Data variation on odd edge, data sampling on even edge | R/W |
| b1 | CPOL | RSPCK Polarity Setting | 0: RSPCK is low when idle<br>1: RSPCK is high when idle | R/W |
| b3, b2 | BRDV[1:0] | Bit Rate Division Setting | b3 b2<br>0 0: These bits select the base bit rate<br>0 1: These bits select the base bit rate divided by 2<br>1 0: These bits select the base bit rate divided by 4<br>1 1: These bits select the base bit rate divided by 8 | R/W |
| b6 to b4 | SSLA[2:0] | SSL Signal Assertion Setting | b6 b4<br>0 0 0: SSL0<br>0 0 1: SSL1<br>0 1 0: SSL2<br>0 1 1: SSL3<br>1 x x: Setting prohibited<br>x: Don't care | R/W |
| b7 | SSLKP | SSL Signal Level Keeping | 0: Negates all SSL signals upon completion of transfer<br>1: Keeps the SSL signal level from the end of transfer until the beginning of the next access | R/W |
| b11 to b8 | SPB[3:0] | RSPI Data Length Setting | b11 b8<br>0100 to 0111: 8 bits<br>1 0 0 0: 9 bits<br>1 0 0 1: 10 bits<br>1 0 1 0: 11 bits<br>1 0 1 1: 12 bits<br>1 1 0 0: 13 bits<br>1 1 0 1: 14 bits<br>1 1 1 0: 15 bits<br>1 1 1 1: 16 bits<br>0 0 0 0: 20 bits<br>0 0 0 1: 24 bits<br>0010, 0011: 32 bits | R/W |
| b12 | LSBF | RSPI LSB First | 0: MSB first<br>1: LSB first | R/W |
| b13 | SPNDEN | RSPI Next-Access Delay Enable | 0: A next-access delay of 1 RSPCK + 2 PCLK<br>1: A next-access delay is equal to the setting of the RSPI next-access delay register (SPND) | R/W |
| b14 | SLNDEN | SSL Negation Delay Setting Enable | 0: An SSL negation delay of 1 RSPCK<br>1: An SSL negation delay is equal to the setting of the RSPI slave select negation delay register (SSLND) | R/W |
| b15 | SCKDEN | RSPCK Delay Setting Enable | 0: An RSPCK delay of 1 RSPCK<br>1: An RSPCK delay is equal to the setting of the RSPI clock delay register (SPCKD) | R/W |

SPCMDm register is used to set a transfer format for the RSPI in master mode. Each channel has eight RSPI command registers (SPCMD0 to SPCMD7). Some of the bits in SPCMD0 register is used to set a transfer mode for the RSPI in slave mode. The RSPI in master mode sequentially references SPCMDm register according to the settings in the SPSCR.SPSLN[2:0] bits, and executes the serial transfer that is set in the referenced SPCMDm register.
SPCMDm register should be set while the transmit buffer is empty (data for the next transfer is not set) and before setting of the data that is to be transmitted when that SPCMDm register is referenced.
SPCMDm that is referenced by the RSPI in master mode can be checked by means of the SPSSR.SPCP[2:0] bits. If the contents of SPCMDm are changed while the SPCR.MSTR bit is 0 and the SPCR.SPE bit is 1, subsequent operations should not be performed.

#### CPHA Bit (RSPCK Phase Setting)

The CPHA bit sets an RSPCK phase of the RSPI in master mode or slave mode. Data communications between RSPI modules require the same RSPCK phase setting between the modules.

#### CPOL Bit (RSPCK Polarity Setting)

The CPOL bit sets an RSPCK polarity of the RSPI in master mode or slave mode. Data communications between RSPI modules require the same RSPCK polarity setting between the modules.

#### BRDV[1:0] Bits (Bit Rate Division Setting)

The BRDV[1:0] bits are used to determine the bit rate. A bit rate is determined by combinations of the settings in the BRDV[1:0] bits and SPBR (refer to section 28.2.8, RSPI Bit Rate Register (SPBR)). The settings in SPBR determine the base bit rate. The settings in the BRDV[1:0] bits are used to select a bit rate which is obtained by dividing the base bit rate by 1, 2, 4, or 8. In SPCMDm register, different BRDV[1:0] bit settings can be specified. This permits the execution of serial transfers at a different bit rate for each command.

#### SSLA[2:0] Bits (SSL Signal Assertion Setting)

The SSLA[2:0] bits control the SSLAi signal assertion when the RSPI performs serial transfers in master mode.
Setting the SSLA[2:0] bits controls the assertion for the SSLAi signal. When an SSLAi signal is asserted, its polarity is determined by the set value in the corresponding SSLP. When the SSLA[2:0] bits are set to 000b in multi-master mode, serial transfers are performed with all the SSL signals in the negated state (as the SSLA0 pin acts as input).
When using the RSPI in slave mode, set the SSLA[2:0] bits to 000b.

#### SSLKP Bit (SSL Signal Level Keeping)

When the RSPI in master mode performs a serial transfer, the SSLKP bit specifies whether the SSLAi signal level for the current command is to be kept or negated between the SSL negation timing associated with the current command and the SSL assertion timing associated with the next command.
Setting the SSLKP bit to 1 enables a burst transfer. For details, refer to section 28.3.10.1, Master Mode Operation (4) Burst Transfer.
When using the RSPI in slave mode, the SSLKP bit should be set to 0.

#### SPB[3:0] Bits (RSPI Data Length Setting)

The SPB[3:0] bits set a transfer data length for the RSPI in master mode or slave mode.

#### LSBF Bit (RSPI LSB First)

The LSBF bit sets the data format of the RSPI in master mode or slave mode to MSB first or LSB first.

### SPNDEN Bit (RSPI Next-Access Delay Enable)

The SPNDEN bit sets the period from the time the RSPI in master mode terminates a serial transfer and sets the SSLAi signal inactive until the RSPI enables the SSLAi signal assertion for the next access (next-access delay). If the SPNDEN bit is 0, the RSPI sets the next-access delay to 1 RSPCK + 2 PCLK. If the SPNDEN bit is 1, the RSPI inserts a next-access delay in compliance with the SPND setting.

When using the RSPI in slave mode, the SPNDEN bit should be set to 0.

### SLNDEN Bit (SSL Negation Delay Setting Enable)

The SLNDEN bit sets the period from the time the RSPI in master mode stops RSPCK oscillation until the RSPI sets the SSLAi signal inactive (SSL negation delay). If the SLNDEN bit is 0, the RSPI sets the SSL negation delay to 1 RSPCK. If the SLNDEN bit is 1, the RSPI negates the SSL signal at an SSL negation delay in compliance with the SSLND setting.

When using the RSPI in slave mode, the SLNDEN bit should be set to 0.

### SCKDEN Bit (RSPCK Delay Setting Enable)

The SCKDEN bit sets the period from the point when the RSPI in master mode activates the SSLAi signal until the RSPCK starts oscillation (RSPI clock delay). If the SCKDEN bit is 0, the RSPI sets the RSPCK delay to 1 RSPCK. If the SCKDEN bit is 1, the RSPI starts the oscillation of RSPCK at an RSPCK delay in compliance with the SPCKD setting.

When using the RSPI in slave mode, the SCKDEN bit should be set to 0.

## 28.3 Operation

In this section, the serial transfer period means a period from the beginning of driving valid data to the fetching of the final valid data.

### 28.3.1 Overview of RSPI Operations

The RSPI is capable of synchronous serial transfers in slave mode (SPI operation), single-master mode (SPI operation), multi-master mode (SPI operation), slave mode (clock synchronous operation), and master mode (clock synchronous operation). A particular mode of the RSPI can be selected by using the MSTR, MODFEN, and SPMS bits in SPCR. Table 28.5 lists the relationship between RSPI modes and SPCR settings, and a description of each mode.

**Table 28.5 Relationship between RSPI Modes and SPCR Settings and Description of Each Mode**

| Mode | Slave (SPI Operation) | Single-Master (SPI Operation) | Multi-Master (SPI Operation) | Slave (Clock Synchronous Operation) | Master (Clock Synchronous Operation) |
|---|---|---|---|---|---|
| MSTR bit setting | 0 | 1 | 1 | 0 | 1 |
| MODFEN bit setting | 0 or 1 | 0 | 1 | 0 | 0 |
| SPMS bit setting | 0 | 0 | 0 | 1 | 1 |
| RSPCKA signal | Input | Output | Output/Hi-Z | Input | Output |
| MOSIA signal | Input | Output | Output/Hi-Z | Input | Output |
| MISOA signal | Output/Hi-Z | Input | Input | Output | Input |
| SSLA0 signal | Input | Output | Input | Hi-Z*1 | Hi-Z*1 |
| SSLA1 to SSLA3 signals | Hi-Z*1 | Output | Output/Hi-Z | Hi-Z*1 | Hi-Z*1 |
| SSL polarity modification function | Supported | Supported | Supported | — | — |
| Transfer rate | Up to PCLK/8 | Up to PCLK/2 | Up to PCLK/2 | Up to PCLK/8 | Up to PCLK/2 |
| Clock source | RSPCK input | On-chip baud rate generator | On-chip baud rate generator | RSPCK input | On-chip baud rate generator |
| Clock polarity | Two | | | | |
| Clock phase | Two | Two | Two | One (CPHA = 1) | Two |
| First transfer bit | MSB/LSB | | | | |
| Transfer data length | 8 to 16, 20, 24, 32 bits | | | | |
| Burst transfer | Possible (CPHA = 1) | Possible (CPHA = 0,1) | Possible (CPHA = 0,1) | — | — |
| RSPCK delay control | Not supported | Supported | Supported | Not supported | Supported |
| SSL negation delay control | Not supported | Supported | Supported | Not supported | Supported |
| Next-access delay control | Not supported | Supported | Supported | Not supported | Supported |
| Transfer activation method | SSL input active or RSPCK oscillation | Transmit buffer is written to at generation of a transmit buffer empty interrupt request | Transmit buffer is written to at generation of a transmit buffer empty interrupt request | RSPCK oscillation | Transmit buffer is written to at generation of a transmit buffer empty interrupt request |
| Sequence control | Not supported | Supported | Supported | Not supported | Supported |
| Transmit buffer empty detection | Supported | | | | |
| Receive buffer full detection | Supported*2 | | | | |
| Overrun error detection | Supported*2 | | | | |
| Parity error detection | Supported*2,*3 | | | | |
| Mode fault error detection | Supported (MODFEN = 1) | Not supported | Supported | Not supported | Not supported |

Note 1. This function is not supported in this mode.
Note 2. When the SPCR.TXMD bit is 1, receiver buffer full detection, overrun error detection, and parity error detection are not performed.
Note 3. When the SPCR2.SPPE bit is 0, parity error detection is not performed.

### 28.3.2 Controlling RSPI Pins

According to the MSTR, MODFEN, and SPMS bits in SPCR and the ODRn.Bi bit for I/O ports, the RSPI can switch pin states. Table 28.6 lists the relationship between pin states and bit settings. Setting the ODRn.Bi bit for an I/O port to 0 selects CMOS output; setting it to 1 selects open-drain output. The I/O port settings should follow this relationship.

**Table 28.6 Relationship between Pin States and Bit Settings**

| Mode | Pin | Pin State*2 | |
|---|---|---|---|
| | | ODRn.Bi Bit for I/O Ports = 0 | ODRn.Bi Bit for I/O Ports = 1 |
| Single-master mode (SPI operation) (MSTR = 1, MODFEN = 0, SPMS = 0) | RSPCKA | CMOS output | Open-drain output |
| | SSLA0 to SSLA3 | CMOS output | Open-drain output |
| | MOSIA | CMOS output | Open-drain output |
| | MISOA | Input | Input |
| Multi-master mode (SPI operation) (MSTR = 1, MODFEN = 1, SPMS = 0) | RSPCKA*3 | CMOS output/Hi-Z | Open-drain output/Hi-Z |
| | SSLA0 | Input | Input |
| | SSLA1 to SSLA3*3 | CMOS output/Hi-Z | Open-drain output/Hi-Z |
| | MOSIA*3 | CMOS output/Hi-Z | Open-drain output/Hi-Z |
| | MISOA | Input | Input |
| Slave mode (SPI operation) (MSTR = 0, SPMS = 0) | RSPCKA | Input | Input |
| | SSLA0 | Input | Input |
| | SSLA1 to SSLA3*5 | Hi-Z*1 | Hi-Z*1 |
| | MOSIA | Input | Input |
| | MISOA*4 | CMOS output/Hi-Z | Open-drain output/Hi-Z |
| Master mode (Clock synchronous operation) (MSTR = 1, MODFEN = 0, SPMS = 1) | RSPCKA | CMOS output | Open-drain output |
| | SSLA0 to SSLA3*5 | Hi-Z*1 | Hi-Z*1 |
| | MOSIA | CMOS output | Open-drain output |
| | MISOA | Input | Input |
| Slave mode (Clock synchronous operation) (MSTR = 0, SPMS = 1) | RSPCKA | Input | Input |
| | SSLA0 to SSLA3*5 | Hi-Z*1 | Hi-Z*1 |
| | MOSIA | Input | Input |
| | MISOA | CMOS output | Open-drain output |

Note 1. This function is not supported in this mode.
Note 2. RSPI settings are not reflected in the multiplex pins for which the RSPI function is not selected.
Note 3. When SSLA0 is at the active level, the pin state is Hi-Z.
Note 4. When SSLA0 is at the non-active level or the SPCR.SPE bit is cleared (= 0), the pin state is Hi-Z.
Note 5. These pins are available for use as I/O port pins.

The RSPI in single-master mode (SPI operation) or multi-master mode (SPI operation) determines MOSI signal values during the SSL negation period (including the SSL retention period during a burst transfer) according to MOIFE and MOIFV bit settings in SPPCR, as listed in Table 28.7.

**Table 28.7 MOSI Signal Value Determination during SSL Negation Period**

| MOIFE Bit | MOIFV Bit | MOSIA Signal Value during SSL Negation Period |
|---|---|---|
| 0 | 0, 1 | Final data from previous transfer |
| 1 | 0 | Always low |
| 1 | 1 | Always High |

## 28.3.3 RSPI System Configuration Examples

### 28.3.3.1 Single Master/Single Slave (with This MCU Acting as Master)

Figure 28.5 shows a single-master/single-slave RSPI system configuration example when this MCU is used as a master. In the single-master/single-slave configuration, the SSLA0 to SSLA3 output of this MCU (master) are not used. The SSL input of the SPI slave is fixed to the low level, and the SPI slave is always maintained in a select state.*1
This MCU (master) always drives the RSPCKA and MOSIA. The SPI slave always drives the MISO.

Note 1. In the transfer format corresponding to the case where the SPCMDm.CPHA bit is 0, there are slave devices for which the SSL signal cannot be fixed to the active level. In situations where the SSL signal cannot be fixed, the SSLAi output of this MCU should be connected to the SSL input of the slave device.

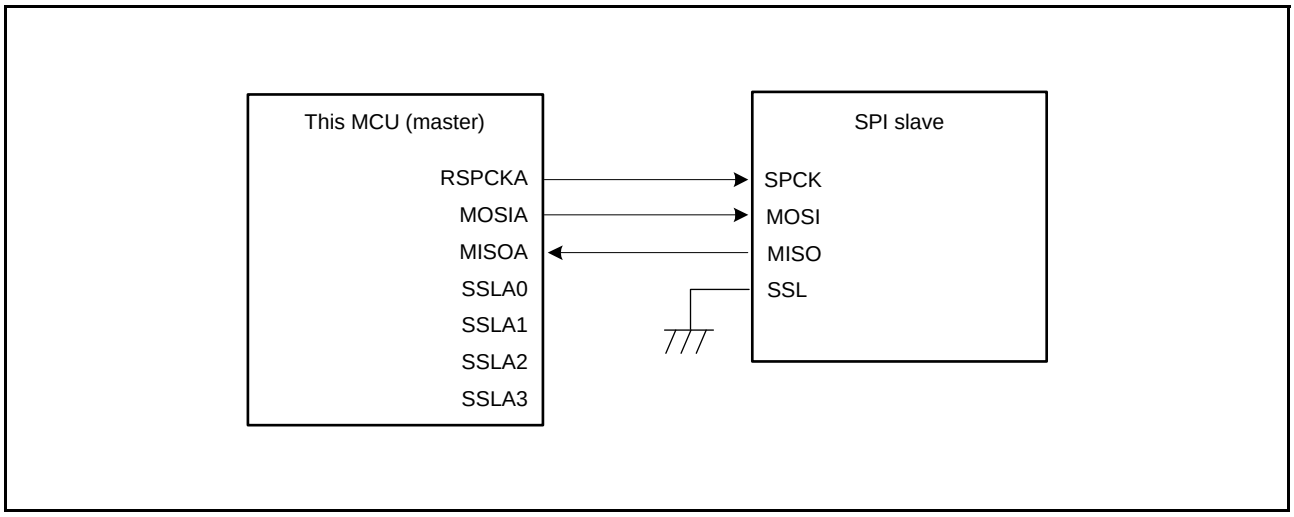



**Figure 28.5 Single-Master/Single-Slave Configuration Example (This MCU = Master)**

### 28.3.3.2 Single Master/Single Slave (with This MCU Acting as Slave)

Figure 28.6 shows a single-master/single-slave RSPI system configuration example when this MCU is used as a slave. When this MCU is to operate as a slave, the SSLA0 pin is used as SSL input. The SPI master always drives the RSPCK and MOSI. This MCU (slave) always drives the MISOA.*1

In the single-slave configuration in which the SPCMDm.CPHA bit is set to 1, the SSLA0 input of this MCU (slave) is fixed to the low level, this MCU (slave) is always maintained in a select state, and in this manner it is possible to execute serial transfer (Figure 28.7).

Note 1. When SSLA0 is at the non-active level, the pin state is Hi-Z.

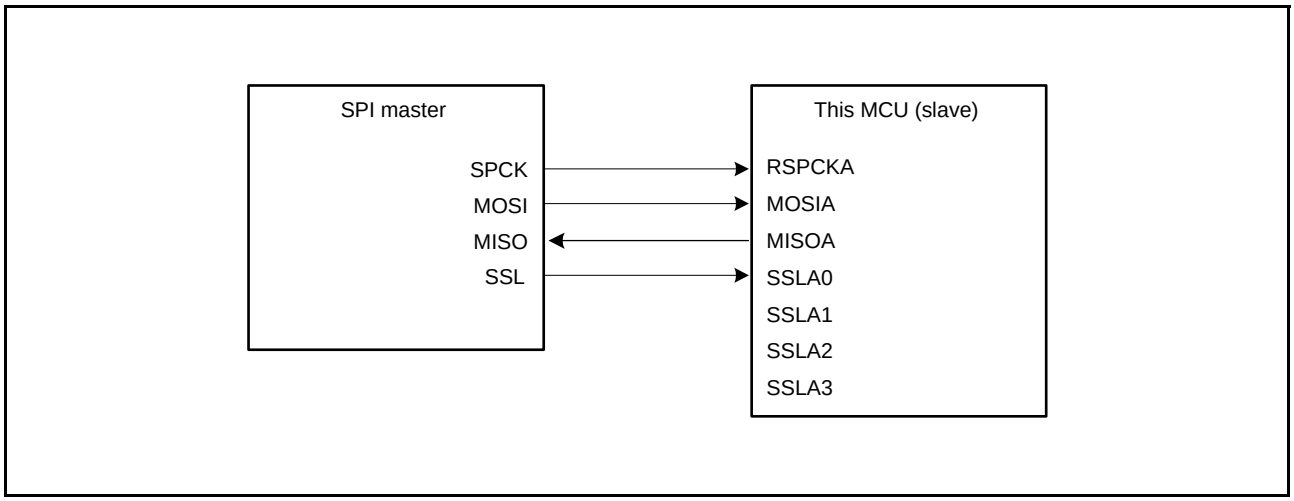



**Figure 28.6 Single-Master/Single-Slave Configuration Example (This MCU = Slave, CPHA = 0)**

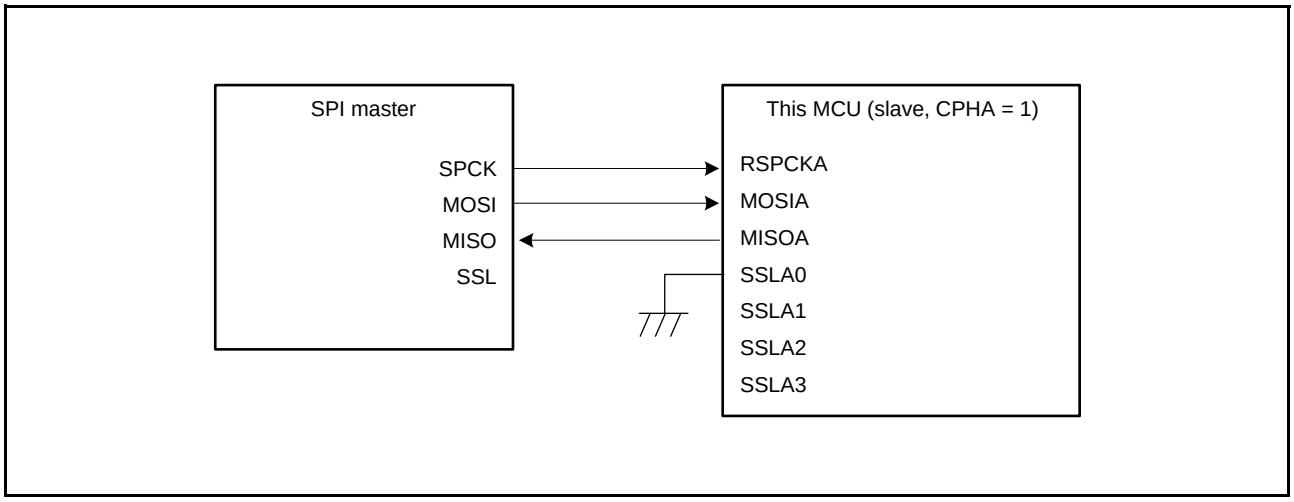



**Figure 28.7 Single-Master/Single-Slave Configuration Example (This MCU = Slave, CPHA = 1)**

### 28.3.3.3 Single Master/Multi-Slave (with This MCU Acting as Master)

Figure 28.8 shows a single-master/multi-slave RSPI system configuration example when this MCU is used as a master. In the example of Figure 28.8, the RSPI system is comprised of this MCU (master) and four slaves (SPI slave 0 to SPI slave 3).

The RSPCKA and MOSIA outputs of this MCU (master) are connected to the RSPCK and MOSI inputs of SPI slave 0 to SPI slave 3. The MISO outputs of SPI slave 0 to SPI slave 3 are all connected to the MISOA input of this MCU (master). SSLA0 to SSLA3 outputs of this MCU (master) are connected to the SSL inputs of SPI slave 0 to SPI slave 3, respectively.

This MCU (master) always drives RSPCK, MOSI, and SSLA0 to SSLA3. Of the SPI slave 0 to SPI slave 3, the slave that receives low-level input into the SSL input drives MISO.

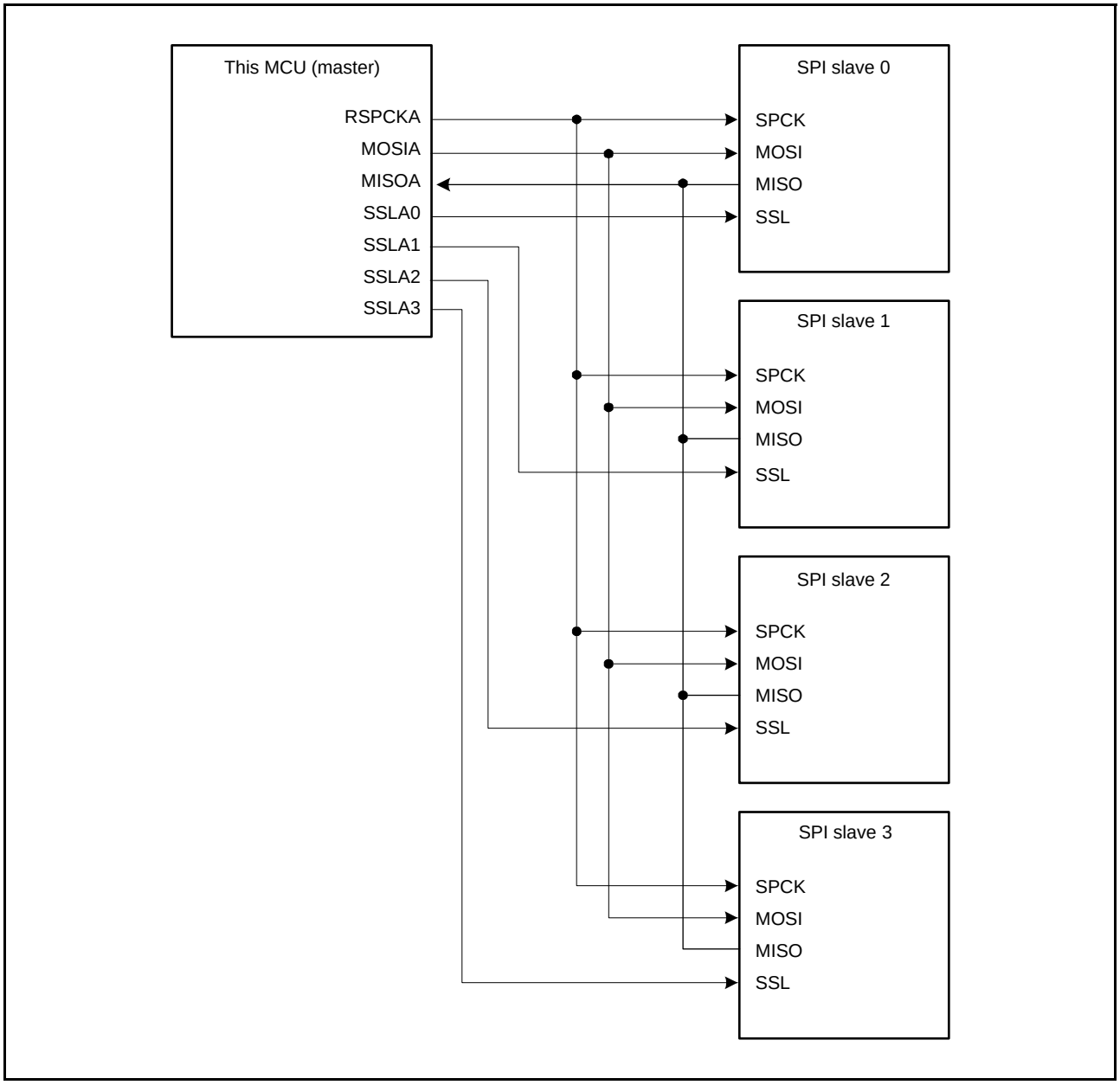



**Figure 28.8 Single-Master/Multi-Slave Configuration Example (This MCU = Master)**

### 28.3.3.4 Single Master/Multi-Slave (with This MCU Acting as Slave)

Figure 28.9 shows a single-master/multi-slave RSPI system configuration example when this MCU is used as a slave. In the example of Figure 28.9, the RSPI system is comprised of an SPI master and two MCUs (slave X and slave Y). The SPCK and MOSI outputs of the SPI master are connected to the RSPCKA and MOSIA inputs of the MCUs (slave X and slave Y). The MISOA outputs of the MCUs (slave X and slave Y) are all connected to the MISO input of the SPI master. SSLX and SSLY outputs of the SPI master are connected to the SSLA0 inputs of the MCUs (slave X and slave Y), respectively.
The SPI master always drives SPCK, MOSI, SSLX, and SSLY. Of the MCUs (slave X and slave Y), the slave that receives low-level input into the SSLA0 input drives MISOA.

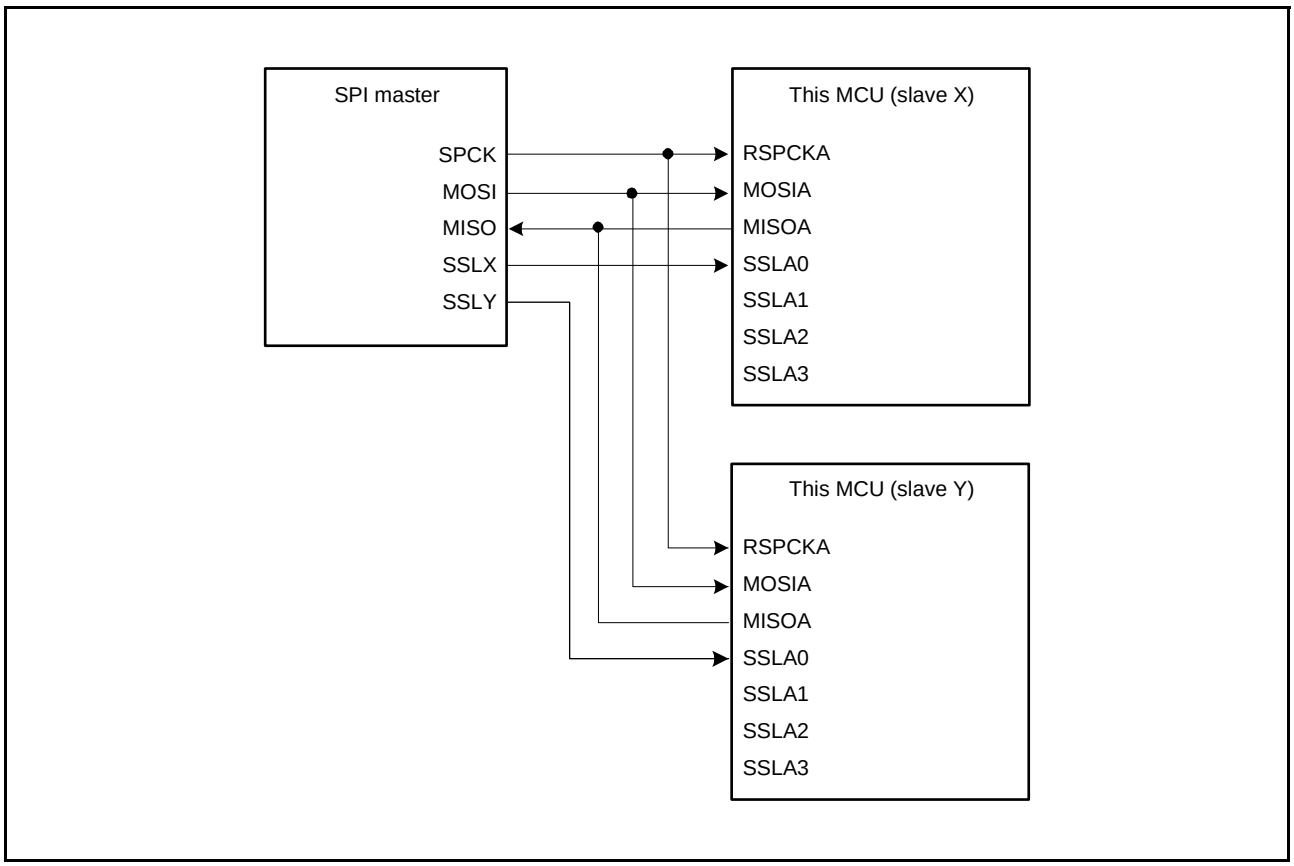



**Figure 28.9 Single-Master/Multi-Slave Configuration Example (This MCU = Slave)**

### 28.3.3.5 Multi-Master/Multi-Slave (with This MCU Acting as Master)

Figure 28.10 shows a multi-master/multi-slave RSPI system configuration example when this MCU is used as a master. In the example of Figure 28.10, the RSPI system is comprised of two MCUs (master X and master Y) and two SPI slaves (SPI slave 1 and SPI slave 2).

The RSPCKA and MOSIA outputs of the MCUs (master X and master Y) are connected to the RSPCK and MOSI inputs of SPI slaves 1 and 2. The MISO outputs of SPI slaves 1 and 2 are connected to the MISOA inputs of the MCUs (master X and master Y). Any generic port Y output from this MCU (master X) is connected to the SSLA0 input of this MCU (master Y). Any generic port X output of this MCU (master Y) is connected to the SSLA0 input of this MCU (master X). The SSLA1 and SSLA2 outputs of the MCUs (master X and master Y) are connected to the SSL inputs of the SPI slaves 1 and 2. In this configuration example, since the system can be comprised solely of SSLA0 input, and SSLA1 and SSLA2 outputs for slave connections, the SSLA3 output of this MCU is not required.

This MCU drives RSPCKA, MOSIA, SSLA1, and SSLA2 when the SSLA0 input level is high. When the SSLA0 input level is low, this MCU detects a mode fault error, sets RSPCKA, MOSIA, SSLA1, and SSLA2 to Hi-Z, and releases the RSPI bus right to the other master. Of the SPI slaves 1 and 2, the slave that receives low-level input into the SSL input drives MISO.

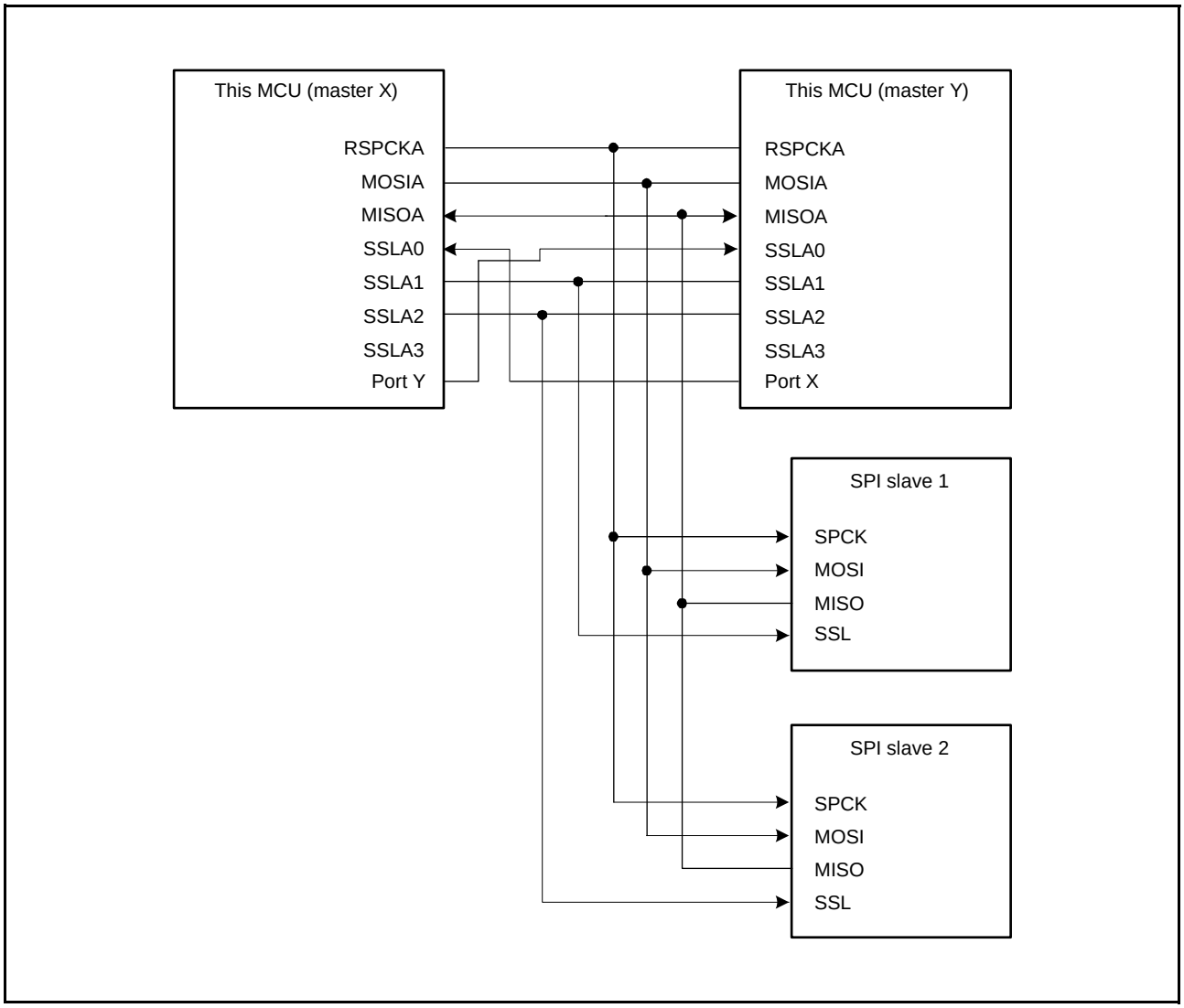



**Figure 28.10 Multi-Master/Multi-Slave Configuration Example (This MCU = Master)**

### 28.3.3.6 Master (Clock Synchronous Operation)/Slave (Clock Synchronous Operation) (with This MCU Acting as Master)

Figure 28.11 shows a master (clock synchronous operation)/slave (clock synchronous operation) RSPI system configuration example when this MCU is used as a master. In the master (clock synchronous operation)/slave (clock synchronous operation) configuration, SSLA0 to SSLA3 of this MCU (master) are not used.
This MCU (master) always drives the RSPCKA and MOSIA. The SPI slave always drives the MISO.

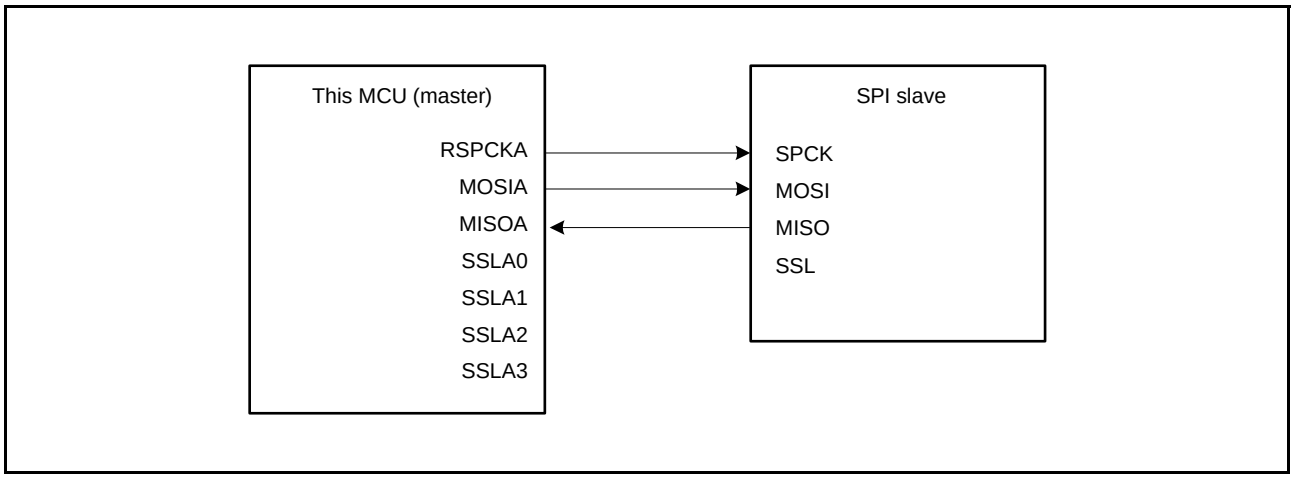



**Figure 28.11 Master (Clock Synchronous Operation)/Slave (Clock Synchronous Operation) Configuration Example (This MCU = Master)**

### 28.3.3.7 Master (Clock Synchronous Operation)/Slave (Clock Synchronous Operation) (with This MCU Acting as Slave)

Figure 28.12 shows a master (clock synchronous operation)/slave (clock synchronous operation) RSPI system configuration example when this MCU is used as a slave. When this MCU is to operate as a slave (clock synchronous operation), this MCU (slave) always drives the MISOA and the SPI master always drives the SPCK and MOSI. In addition, SSLA0 to SSLA3 of this MCU (slave) are not used.
Only in the single-slave configuration in which the SPCMDm.CPHA bit is set to 1, this MCU (slave) can execute serial transfer.

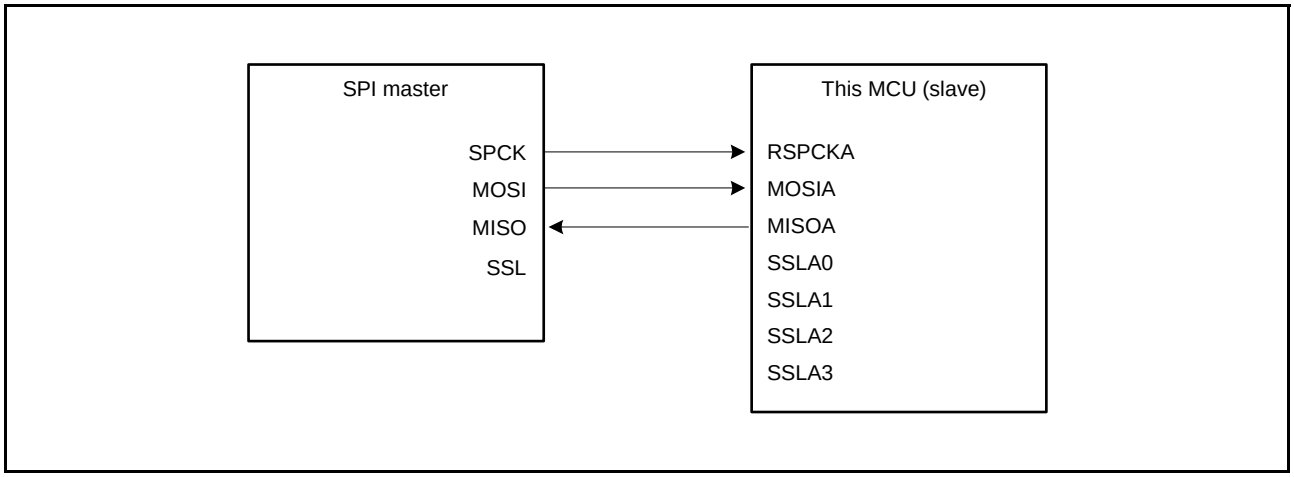



**Figure 28.12 Master (Clock Synchronous Operation)/Slave (Clock Synchronous Operation) Configuration Example (This MCU = Slave, CPHA = 1)**

### 28.3.4 Data Format

The RSPI's data format depends on the settings in RSPI command register m (SPCMDm) (m = 0 to 7) and the parity enable bit of RSPI control register 2 (SPCR2.SPPE). Regardless of whether the MSB or LSB is first, the RSPI treats the range from the LSB bit of the RSPI data register (SPDR) to the selected data length as transfer data.
The format of one frame of data before or after transfer is shown below.

#### (a) With Parity Disabled

When parity is disabled, transmission or reception of data proceeds with the length in bits selected in the RSPI data length setting bits in RSPI command register m (SPCMDm.SPB[3:0]).

#### (b) With Parity Enabled

When parity is enabled, transmission or reception of data proceeds with the length in bits selected in the RSPI data length setting bits in RSPI command register m (SPCMDm.SPB[3:0]). In this case, however, the last bit is a parity bit.

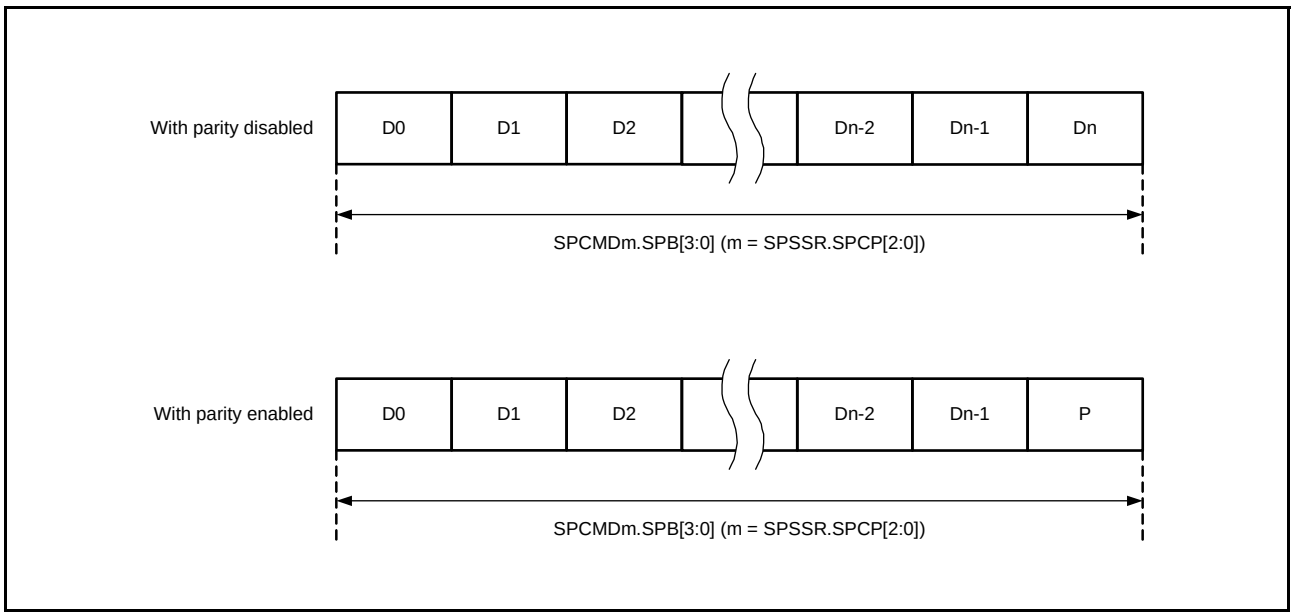



**Figure 28.13 Outline of the Data Format (with Parity Disabled/Enabled)**

### 28.3.4.1 When Parity is Disabled (SPCR2.SPPE = 0)

When parity is disabled, data for transmission are copied to the shift register with no prior processing. A description of the connection between the RSPI data register (SPDR) and the shift register in terms of the combination of MSB or LSB first and data length is given below.

#### (1) MSB First Transfer (32-Bit Data)

Figure 28.14 shows details of operations by the RSPI data register (SPDR) and the shift register in transfer with parity disabled, an RSPI data length of 32 bits, and MSB first selected.
In transmission, bits T31 to T00 from the current stage of the transmit buffer are copied to the shift register. Data for transmission are shifted out from the shift register in order from T31, through T30, and so on to T00.
In reception, received data are shifted in bit by bit through bit 0 of the shift register. When bits R31 to R00 have been collected after input of the required number of cycles of RSPCK, the value in the shift register is copied to the receive buffer.

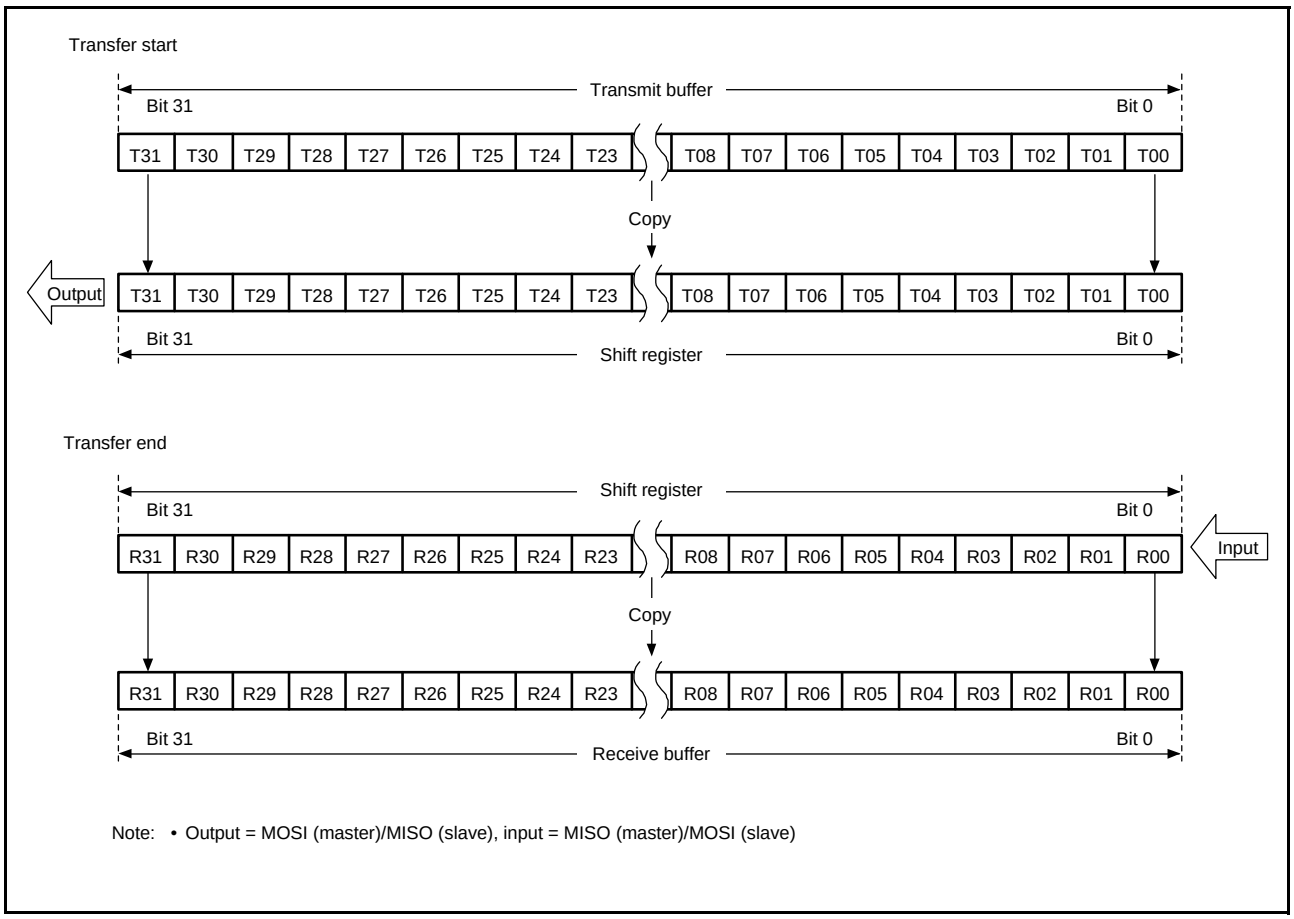



**Figure 28.14 MSB First Transfer (32-Bit Data, Parity Disabled)**

### (2) MSB First Transfer (24-Bit Data)

Figure 28.15 shows details of operations by the RSPI data register (SPDR) and the shift register in transfer with parity disabled, 24 bits as the RSPI data length for an example that is not 32 bits, and MSB first selected.
In transmission, the lower-order 24 bits (T23 to T00) from the current stage of the transmit buffer are copied to the shift register. Data for transmission are shifted out from the shift register in order from T23, through T22, and so on to T00.
In reception, received data are shifted in bit by bit through bit 0 of the shift register. When bits R23 to R00 have been collected after input of the required number of cycles of RSPCK, the value in the shift register is copied to the receive buffer. At this time, the higher-order 8 bits of the transmit buffer are stored in the higher-order 8 bits of the receive buffer.
Writing 0 to bits T31 to T24 at the time of transmission leads to 0 being inserted in the higher-order 8 bits of the receive buffer.

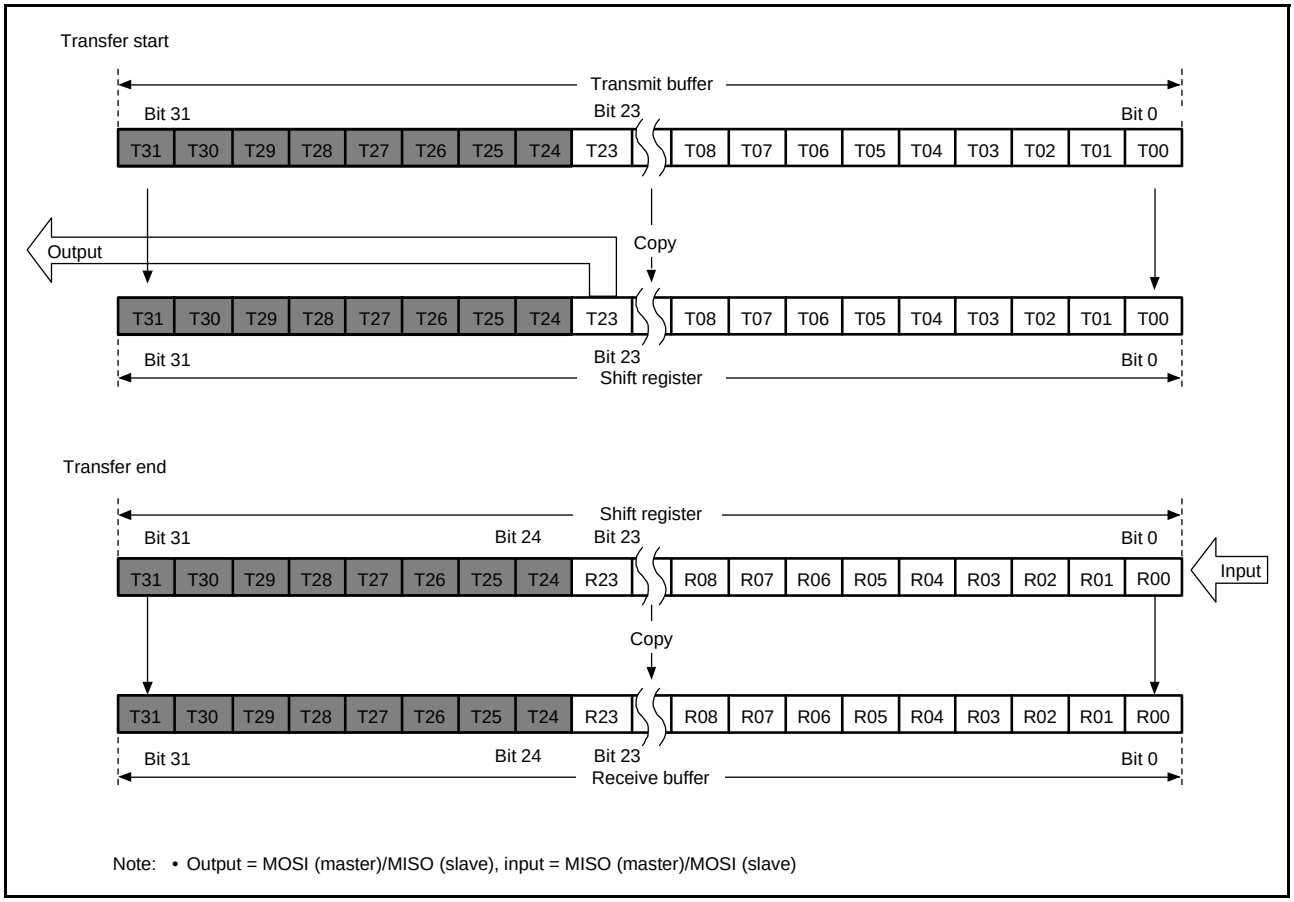



**Figure 28.15 MSB First Transfer (24-Bit Data, Parity Disabled)**

### (3) LSB First Transfer (32-Bit Data)

Figure 28.16 shows details of operations by the RSPI data register (SPDR) and the shift register in transfer with parity disabled, an RSPI data length of 32 bits, and LSB first selected.

In transmission, bits T31 to T00 from the current stage of the transmit buffer are reordered bit by bit to obtain the order T00 to T31 for copying to the shift register. Data for transmission are shifted out from the shift register in order from T00, through T01, and so on to T31.

In reception, received data are shifted in bit by bit through bit 0 of the shift register. When bits R00 to R31 have been collected after input of the required number of cycles of RSPCK, the value in the shift register is copied to the receive buffer.

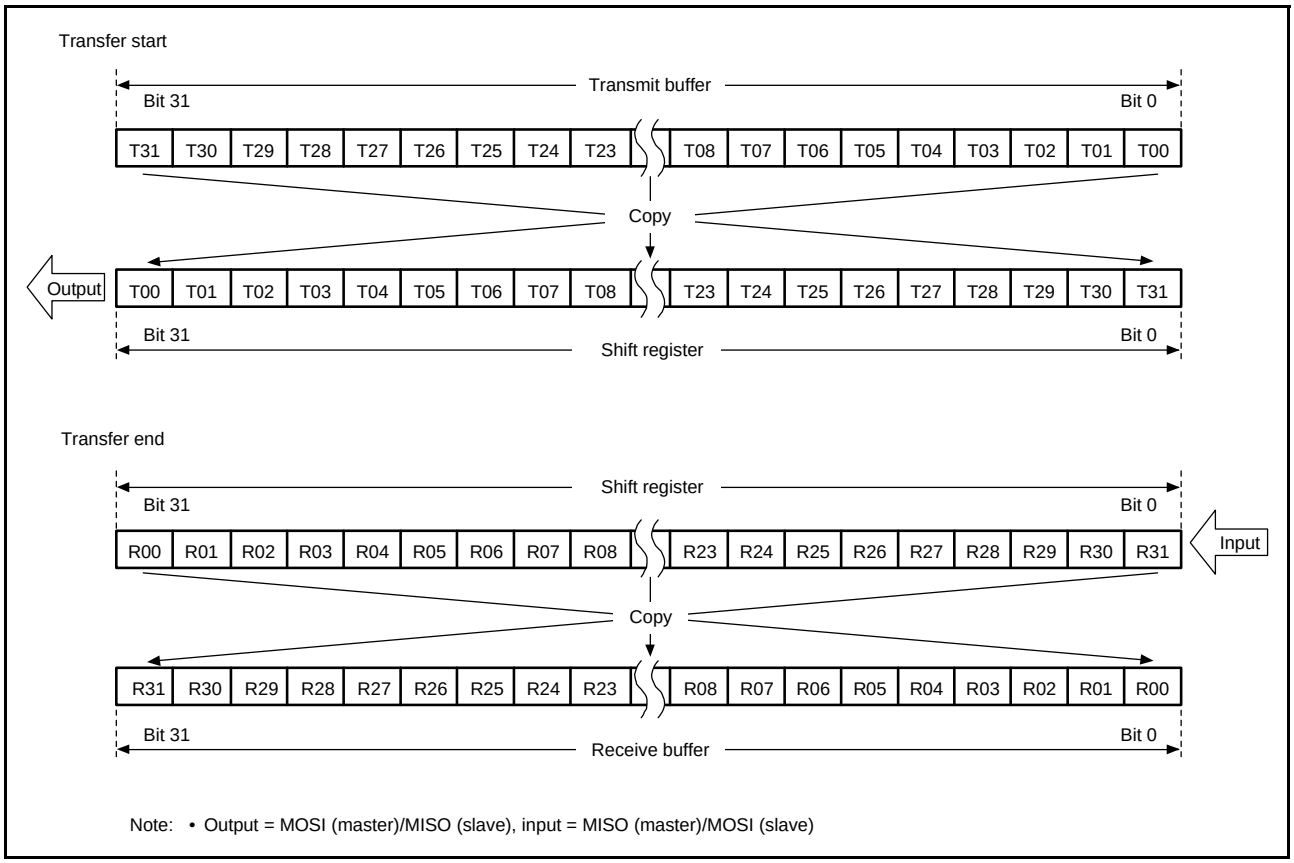



**Figure 28.16 LSB First Transfer (32-Bit Data, Parity Disabled)**

### (4) LSB First Transfer (24-Bit Data)

Figure 28.17 shows details of operations by the RSPI data register (SPDR) and the shift register in transfer with parity disabled, 24 bits as the RSPI data length for an example that is not 32 bits, and LSB first selected.

In transmission, the lower-order 24 bits (T23 to T00) from the current stage of the transmit buffer are reordered bit by bit to obtain the order T00 to T23 for copying to the shift register. Data for transmission are shifted out from the shift register in order from T00, through T01, and so on to T23.

In reception, received data are shifted in bit by bit through bit 8 of the shift register. When bits R00 to R23 have been collected after input of the required number of cycles of RSPCK, the value in the shift register is copied to the receive buffer.

At this time, the higher-order 8 bits of the transmit buffer are stored in the higher-order 8 bits of the receive buffer.

Writing 0 to bits T31 to T24 at the time of transmission leads to 0 being inserted in the higher-order 8 bits of the receive buffer.

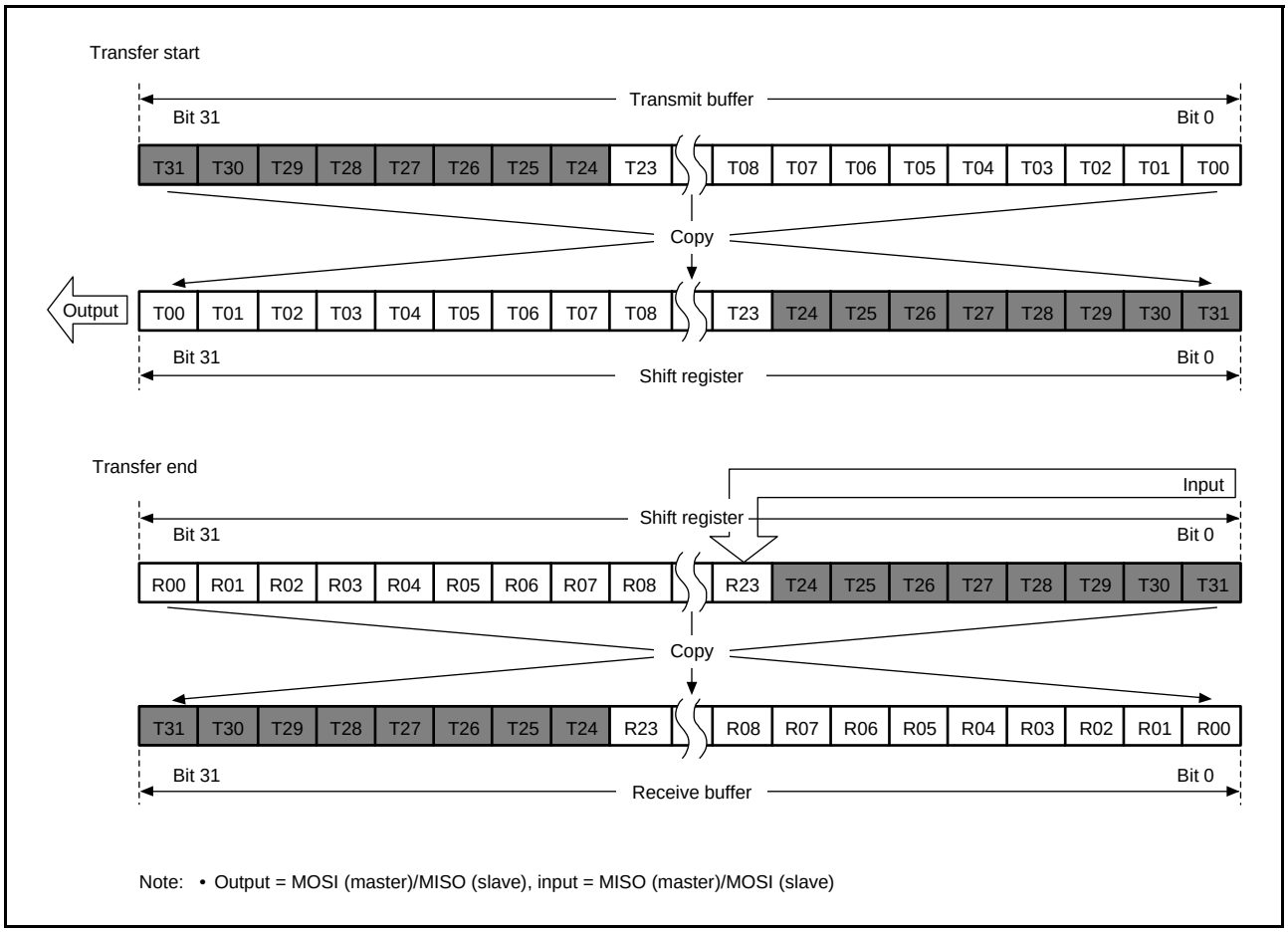



**Figure 28.17 LSB First Transfer (24-Bit Data, Parity Disabled)**

### 28.3.4.2 When Parity is Enabled (SPCR2.SPPE = 1)

When parity is enabled, the lowest-order bit of the data for transmission becomes a parity bit. Hardware calculates the value of the parity bit.

#### (1) MSB First Transfer (32-Bit Data)

Figure 28.18 shows details of operations by the RSPI data register (SPDR) and the shift register in transfer with parity enabled, an RSPI data length of 32 bits, and MSB first selected.
In transmission, the value of the parity bit (P) is calculated from bits T31 to T01. This replaces the final bit, T00, and the whole is copied to the shift register. Data are transmitted in the order T31, T30, …, T01, and P.
In reception, received data are shifted in bit by bit through bit 0 of the shift register. When bits R31 to P have been collected after input of the required number of cycles of RSPCK, the value in the shift register is copied to the receive buffer. On copying of data to the shift register, the data from R31 to P are checked by judging the parity.

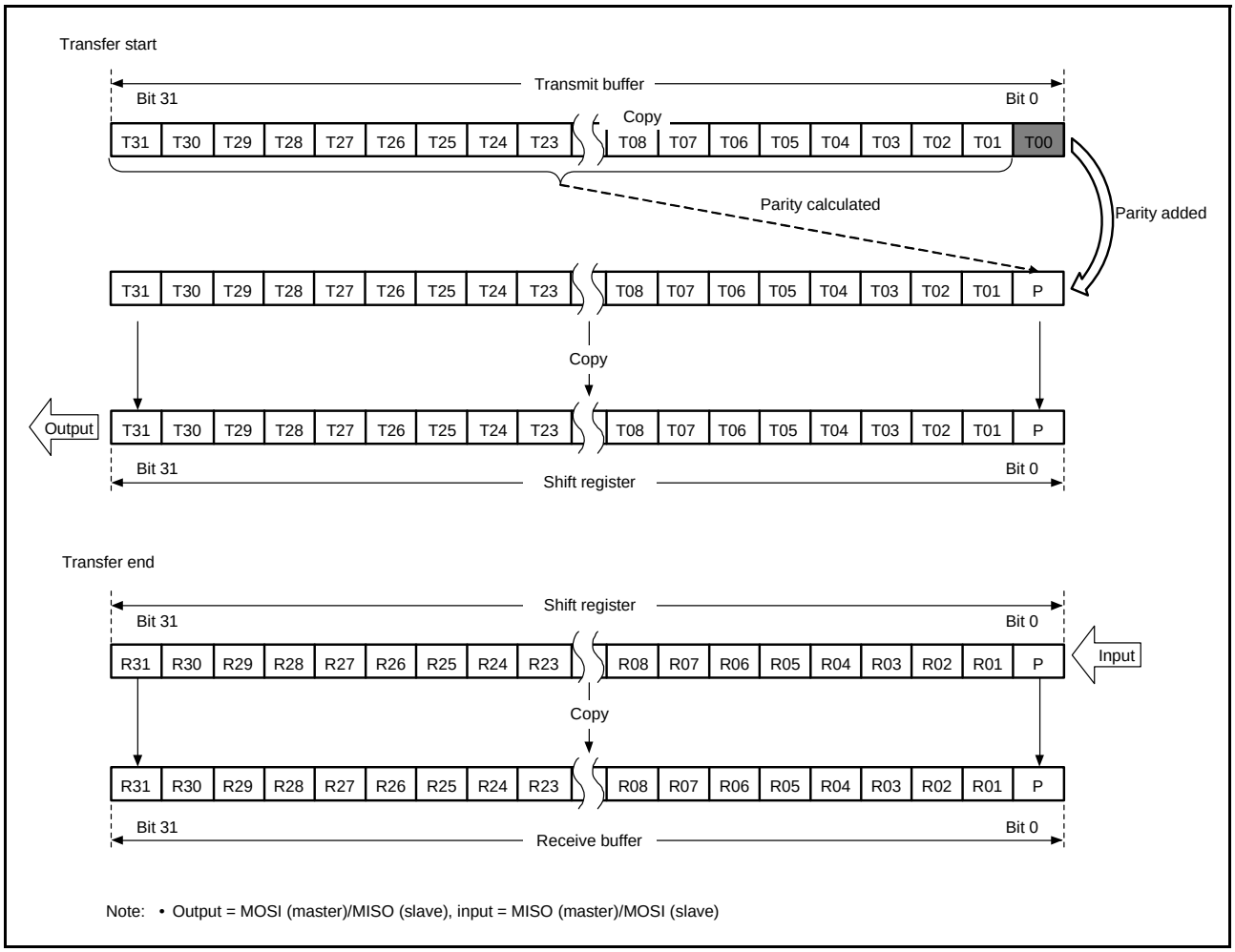



**Figure 28.18 MSB First Transfer (32-Bit Data, Parity Enabled)**

### (2) MSB First Transfer (24-Bit Data)

Figure 28.19 shows details of operations by the RSPI data register (SPDR) and the shift register in transfer with parity enabled, 24 bits as the RSPI data length for an example that is not 32 bits, and MSB first selected.
In transmission, the value of the parity bit (P) is calculated from bits T23 to T01. This replaces the final bit, T00, and the whole is copied to the shift register. Data are transmitted in the order T23, T22, …, T01, and P.
In reception, received data are shifted in bit by bit through bit 0 of the shift register. When bits R23 to P have been collected after input of the required number of cycles of RSPCK, the value in the shift register is copied to the receive buffer. On copying of data to the shift register, the data from R23 to P are checked by judging the parity. At this time, the higher-order 8 bits of the transmit buffer are stored in the higher-order 8 bits of the receive buffer. Writing 0 to bits T31 to T24 at the time of transmission leads to 0 being inserted in the higher-order 8 bits of the receive buffer.

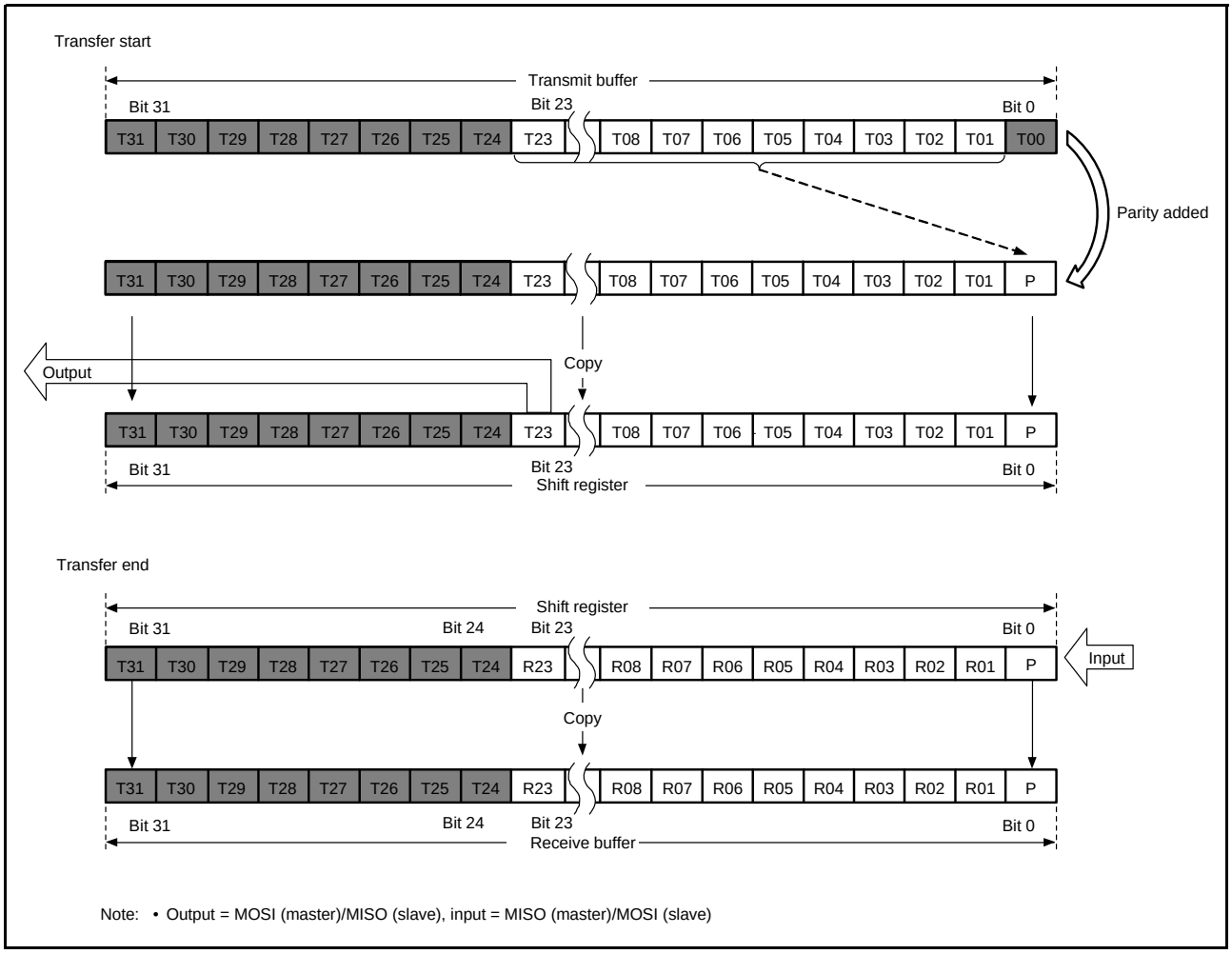



**Figure 28.19 MSB First Transfer (24-Bit Data, Parity Enabled)**

### (3) LSB First Transfer (32-Bit Data)

Figure 28.20 shows details of operations by the RSPI data register (SPDR) and the shift register in transfer with parity enabled, an RSPI data length of 32 bits, and LSB first selected.

In transmission, the value of the parity bit (P) is calculated from bits T30 to T00. This replaces the final bit, T31, and the whole is copied to the shift register. Data are transmitted in the order T00, T01, …, T30, and P.

In reception, received data are shifted in bit by bit through bit 0 of the shift register. When bits R00 to P have been collected after input of the required number of cycles of RSPCK, the value in the shift register is copied to the receive buffer. On copying of data to the shift register, the data from R00 to P are checked by judging the parity.

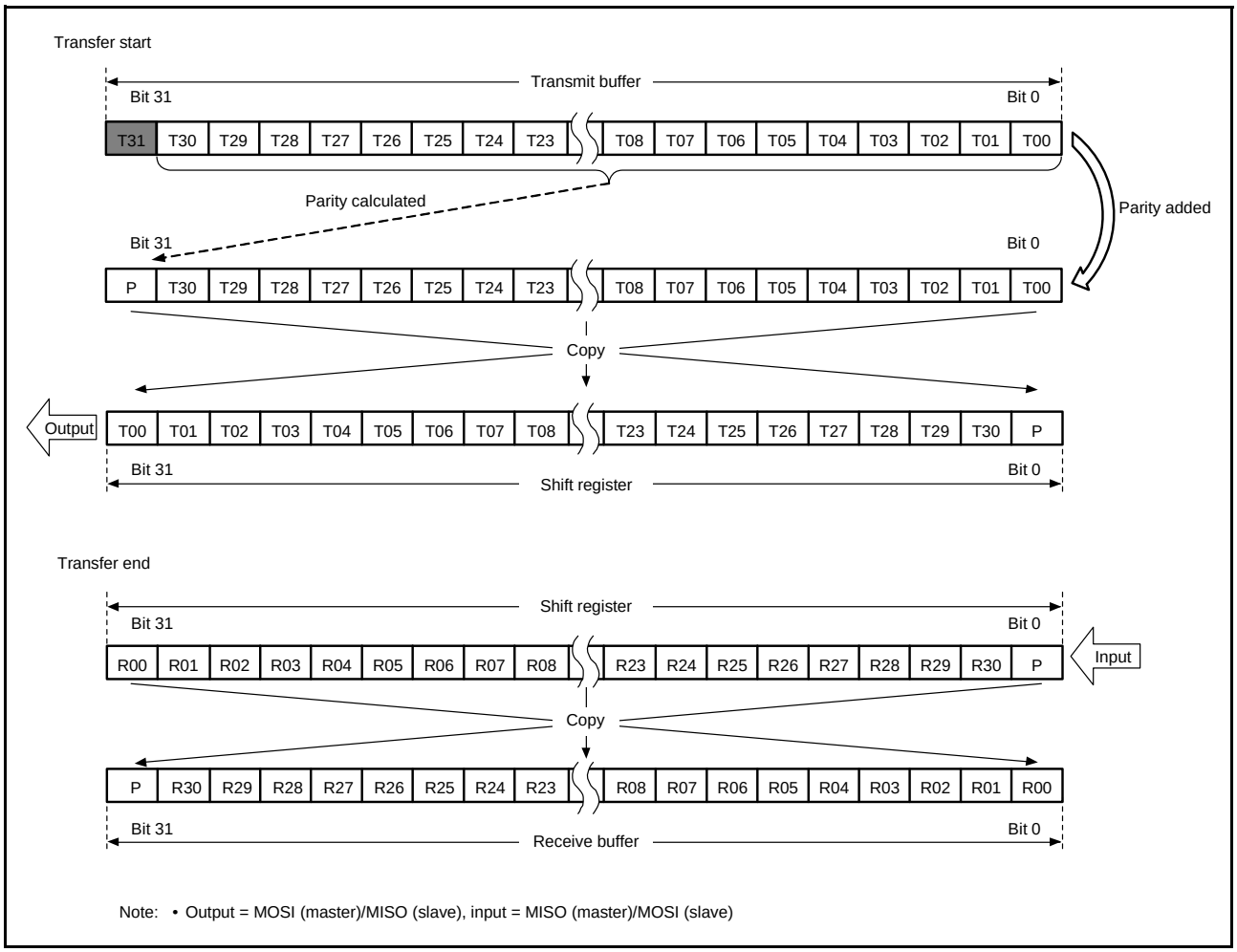



**Figure 28.20 LSB First Transfer (32-Bit Data, Parity Enabled)**

### (4) LSB First Transfer (24-Bit Data)

Figure 28.21 shows details of operations by the RSPI data register (SPDR) and the shift register in transfer with parity enabled, 24 bits as the RSPI data length for an example that is not 32 bits, and LSB first selected.
In transmission, the value of the parity bit (P) is calculated from bits T22 to T00. This replaces the final bit, T23, and the whole is copied to the shift register. Data are transmitted in the order T00, T01, …, T22, and P.
In reception, received data are shifted in bit by bit through bit 8 of the shift register. When bits R00 to P have been collected after input of the required number of cycles of RSPCK, the value in the shift register is copied to the receive buffer. On copying of data to the shift register, the data from R00 to P are checked by judging the parity. At this time, the higher-order 8 bits of the transmit buffer are stored in the higher-order 8 bits of the receive buffer. Writing 0 to bits T31 to T24 at the time of transmission leads to 0 being inserted in the higher-order 8 bits of the receive buffer.

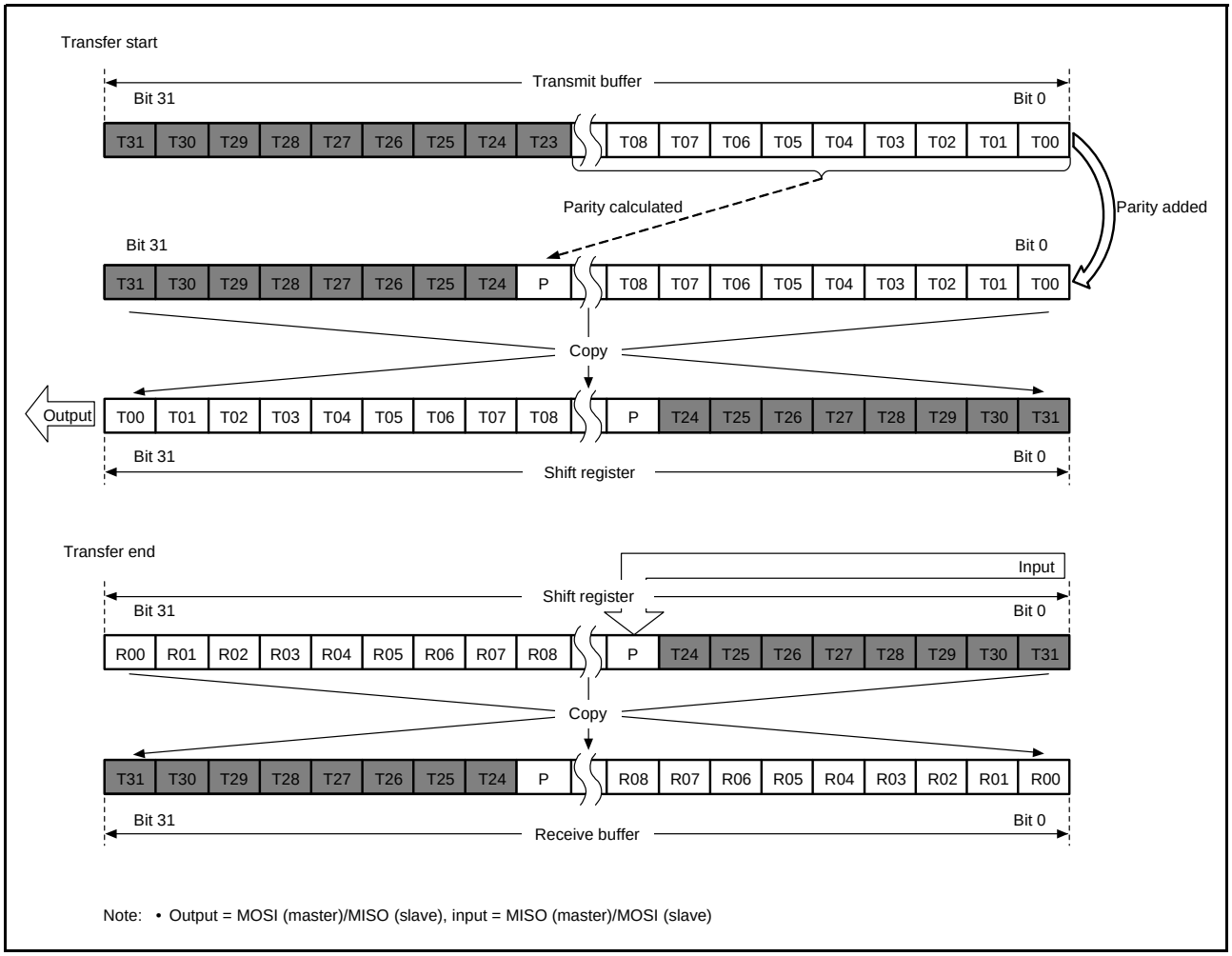



**Figure 28.21 LSB First Transfer (24-Bit Data, Parity Enabled)**

### 28.3.5 Transfer Format

#### 28.3.5.1 CPHA = 0

Figure 28.22 shows a sample transfer format for the serial transfer of 8-bit data when the SPCMDm.CPHA bit is 0. Note that clock synchronous operation (the SPCR.SPMS bit is 1) should not performed when the RSPI operates in slave mode (SPCR.MSTR = 0) and the CPHA bit is 0. In Figure 28.22, RSPCKA (CPOL = 0) indicates the RSPCKA signal waveform when the SPCMDm.CPOL bit is 0; RSPCKA (CPOL = 1) indicates the RSPCKA signal waveform when the CPOL bit is 1. The sampling timing represents the timing at which the RSPI fetches serial transfer data into the shift register. The I/O directions of the signals depend on the RSPI settings. For details, refer to section 28.3.2, Controlling RSPI Pins.

When the SPCMDm.CPHA bit is 0, the driving of valid data to the MOSIA and MISOA signals commences at an SSLAi signal assertion timing. The first RSPCKA signal change timing that occurs after the SSLAi signal assertion becomes the first transfer data fetch timing. After this timing, data is sampled at every 1 RSPCK cycle. The change timing for MOSIA and MISOA signals is always 1/2 RSPCK cycles after the transfer data fetch timing. The CPOL bit setting does not affect the RSPCK signal operation timing; it only affects the signal polarity.

t1 denotes a period from an SSLAi signal assertion to RSPCKA oscillation (RSPCK delay). t2 denotes a period from the termination of RSPCKA oscillation to an SSLAi signal negation (SSL negation delay). t3 denotes a period in which SSLAi signal assertion is suppressed for the next transfer after the end of serial transfer (next-access delay). t1, t2, and t3 are controlled by a master device running on the RSPI system. For a description of t1, t2, and t3 when the RSPI of this MCU is in master mode, refer to section 28.3.10.1, Master Mode Operation.

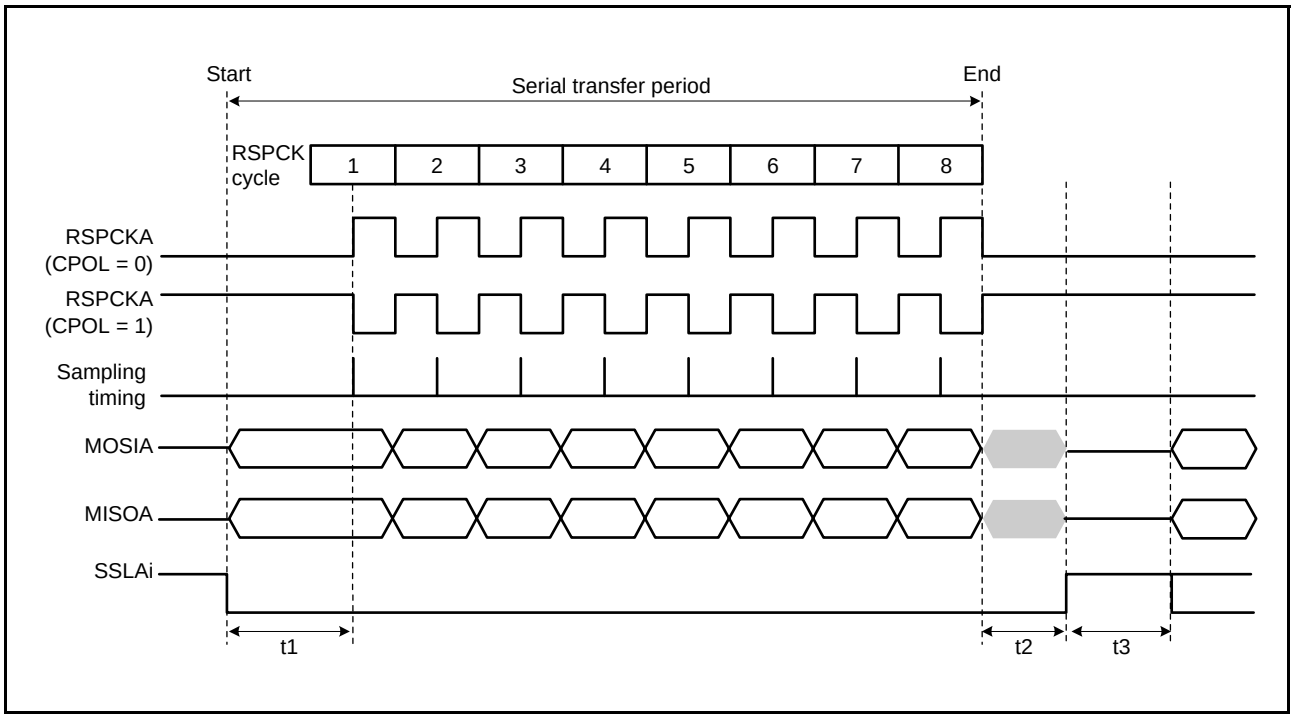



**Figure 28.22 RSPI Transfer Format (CPHA = 0)**

### 28.3.5.2 CPHA = 1

Figure 28.23 shows a sample transfer format for the serial transfer of 8-bit data when the SPCMDm.CPHA bit is 1. However, when the SPCR.SPMS bit is 1, the SSLAi signals are not used, and only the three signals RSPCKA, MOSIA, and MISOA handle communications. In Figure 28.23, RSPCK (CPOL = 0) indicates the RSPCKA signal waveform when the SPCMDm.CPOL bit is 0; RSPCK (CPOL = 1) indicates the RSPCKA signal waveform when the CPOL bit is 1. The sampling timing represents the timing at which the RSPI fetches serial transfer data into the shift register. The I/O directions of the signals depend on the RSPI mode (master or slave). For details, refer to section 28.3.2, Controlling RSPI Pins.

When the SPCMDm.CPHA bit is 1, the driving of invalid data to the MISOA signal commences at an SSLAi signal assertion timing. The output of valid data to the MOSIA and MISOA signals commences at the first RSPCKA signal change timing that occurs after the SSLAi signal assertion. After this timing, data is updated at every 1 RSPCK cycle. The transfer data fetch timing is always 1/2 RSPCK cycles after the data update timing. The SPCMDm.CPOL bit setting does not affect the RSPCKA signal operation timing; it only affects the signal polarity.

t1, t2, and t3 are the same as those in the case of CPHA = 0. For a description of t1, t2, and t3 when the RSPI of this MCU is in master mode, refer to section 28.3.10.1, Master Mode Operation.

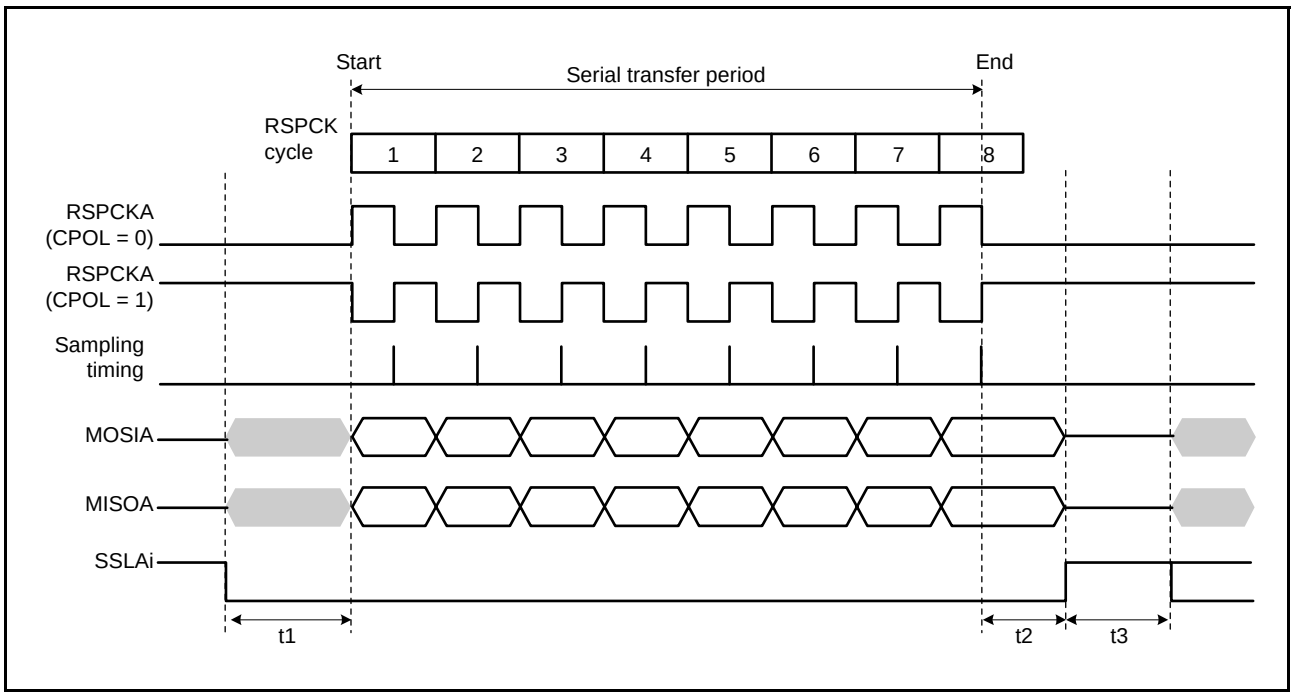



**Figure 28.23 RSPI Transfer Format (CPHA = 1)**

### 28.3.6 Communications Operating Mode

Full-duplex synchronous serial communications or transmit operations only can be selected by the communications operating mode select bit (SPCR.TXMD). The SPDR access shown in Figure 28.24 and Figure 28.25 indicates the condition of access to the SPDR register, where W denotes a write cycle.

#### 28.3.6.1 Full-Duplex Synchronous Serial Communications (SPCR.TXMD = 0)

Figure 28.24 shows an example of operation when the communications operating mode select bit (SPCR.TXMD) is set to 0. In the example in Figure 28.24, the RSPI performs an 8-bit serial transfer in which the SPDCR.SPFC[1:0] bits are 00b, the SPCMDm.CPHA bit is 1, and the SPCMDm.CPOL bit is 0. The numbers given under the RSPCKA waveform represent the number of RSPCK cycles (i.e., the number of transferred bits).

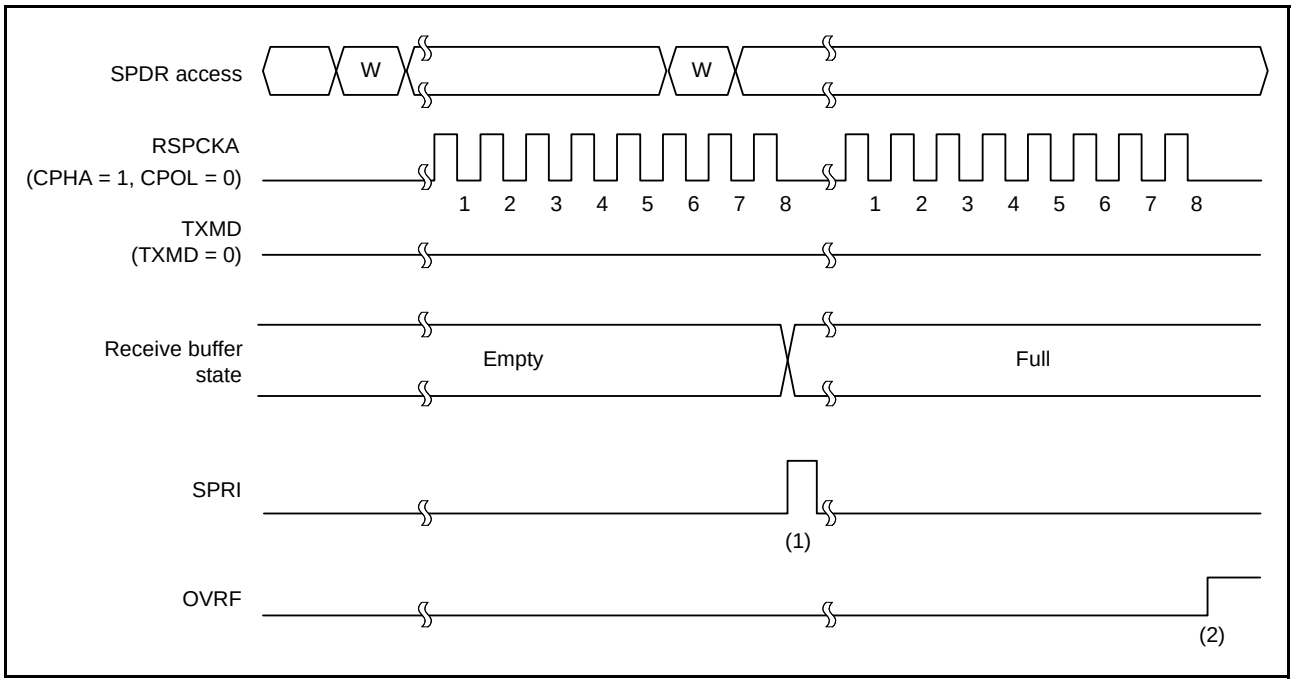



**Figure 28.24 Operation Example of SPCR.TXMD = 0**

The operation of the flags at timings shown in steps (1) and (2) in the figure is described below.

(1) When a serial transfer ends with the receive buffer of SPDR empty, the RSPI generates a receive buffer full interrupt request (SPRI) and copies the received data in the shift register to the receive buffer.

(2) When a serial transfer ends with the receive buffer of SPDR holding data that was received in the previous serial transfer, the RSPI sets the SPSR.OVRF flag to 1 and discards the received data in the shift register.

### 28.3.6.2 Transmit Operations Only (SPCR.TXMD = 1)

Figure 28.25 shows an example of operation when the communications operating mode select bit (SPCR.TXMD) is set to 1. In the example in Figure 28.25, the RSPI performs an 8-bit serial transfer in which the SPDCR.SPFC[1:0] bits are 00b, the SPCMDm.CPHA bit is 1, and the SPCMDm.CPOL bit is 0. The numbers given under the RSPCKA waveform represent the number of RSPCK cycles (i.e., the number of transferred bits).

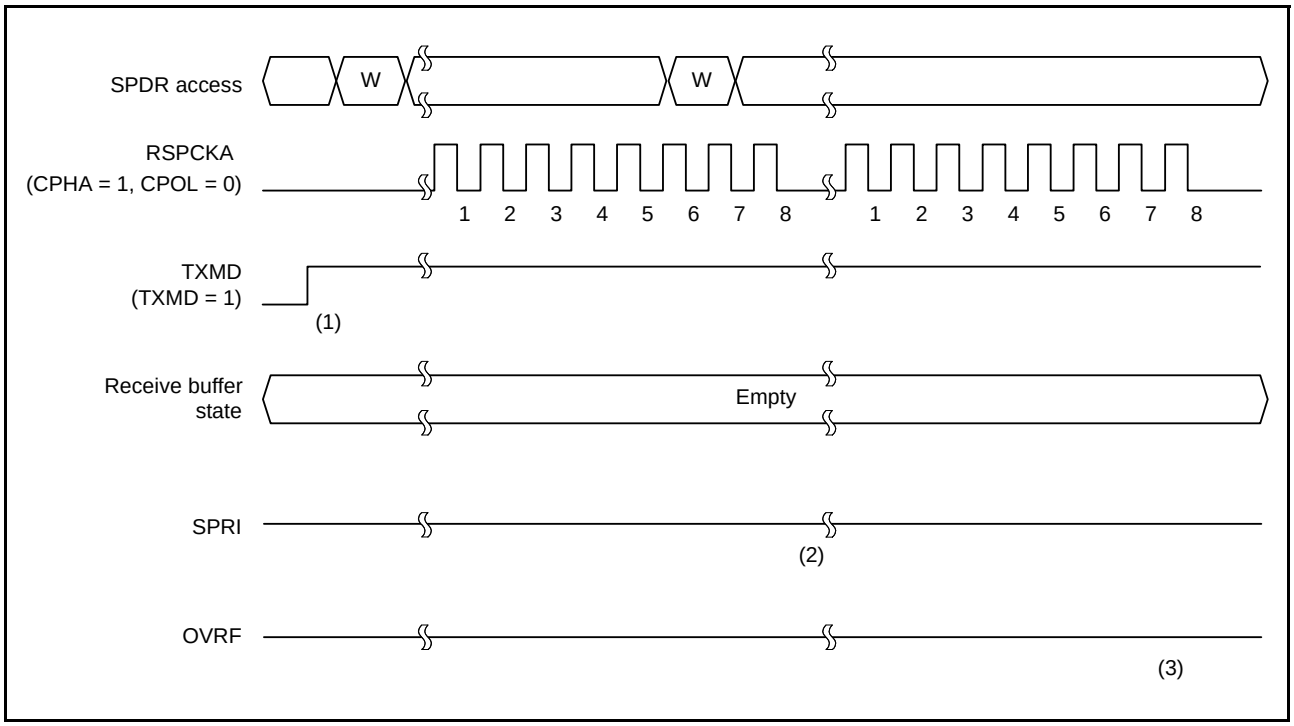



**Figure 28.25 Operation Example of SPCR.TXMD = 1**

The operation of the flags at timings shown in steps (1) to (3) in the figure is described below.

(1) Make sure there is no data left in the receive buffer and the SPSR.OVRF flag is 0 before entering the mode of transmit operations only (SPCR.TXMD = 1).

(2) When a serial transfer ends with the receive buffer of SPDR empty, if the mode of transmit operations only is selected (SPCR.TXMD = 1), the RSPI does not copy the data in the shift register to the receive buffer.

(3) Since the receive buffer of SPDR does not hold data that was received in the previous serial transfer, even when a serial transfer ends, the SPSR.OVRF flag retains the value of 0, and the data in the shift register is not copied to the receive buffer.

When performing transmit operations only (SPCR.TXMD = 1), the RSPI transmits transmit data but does not receive received data. Therefore, the SPSR.OVRF flag remains cleared to 0 at the timings of (1) to (3).

### 28.3.7 Transmit Buffer Empty/Receive Buffer Full Interrupts

Figure 28.26 shows an example of operation of the transmit buffer empty interrupt (SPTI) and the receive buffer full interrupt (SPRI). The SPDR register access shown in Figure 28.26 indicates the condition of access to the SPDR register, where W denotes a write cycle, and R a read cycle. In the example in Figure 28.26, the RSPI performs an 8-bit serial transfer in which the SPCR.TXMD bit is 0, the SPDCR.SPFC[1:0] bits are 00b, the SPCMDm.CPHA bit is 1, and the SPCMDm.CPOL bit is 0. The numbers given under the RSPCKA waveform represent the number of RSPCK cycles (i.e., the number of transferred bits).

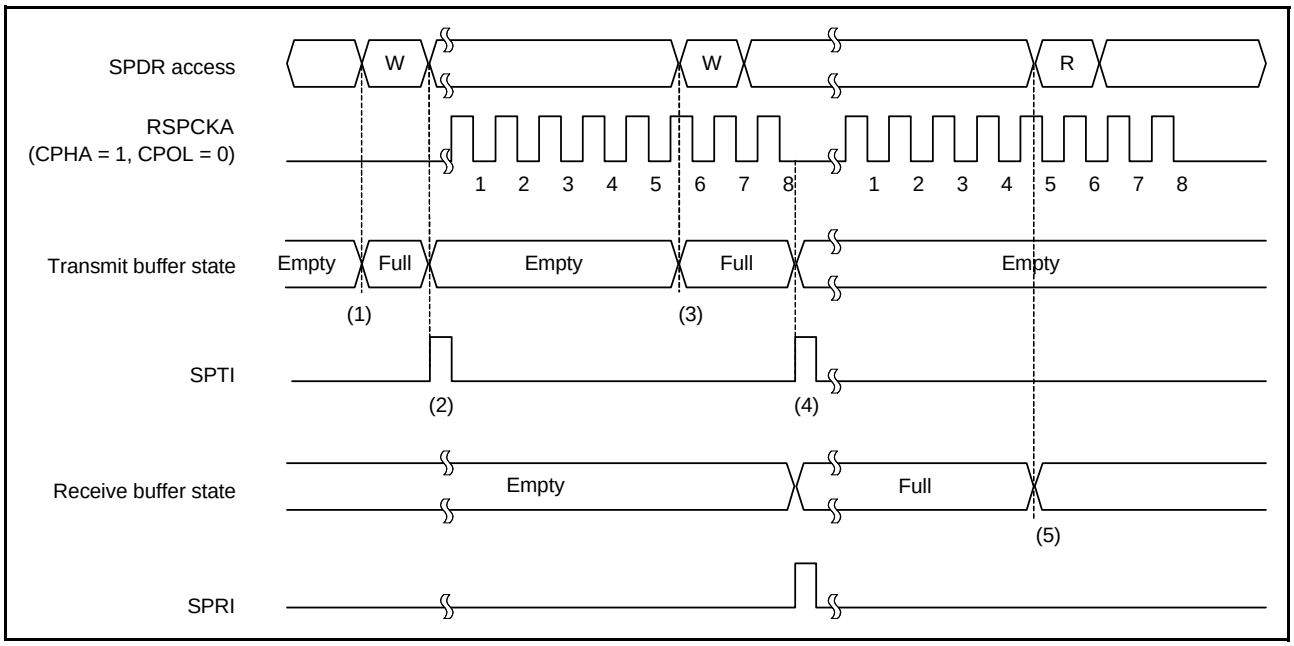



**Figure 28.26 Operation Example of SPTI and SPRI Interrupts**

The operation of the interrupts at timings shown in steps (1) to (5) in the figure is described below.

1. When transmit data is written to SPDR when the transmit buffer of SPDR is empty (data for the next transfer is not set), the RSPI writes data to the transmit buffer.
2. If the shift register is empty, the RSPI copies the data in the transmit buffer to the shift register and generates a transmit buffer empty interrupt request (SPTI). How a serial transfer is started depends on the mode of the RSPI. For details, refer to section 28.3.10, SPI Operation, and section 28.3.11, Clock Synchronous Operation.
3. When transmit data is written to SPDR by the transmit buffer empty interrupt routine, the data is transferred to the transmit buffer. Because the data being transferred serially is stored in the shift register, the RSPI does not copy the data in the transmit buffer to the shift register.
4. When the serial transfer ends with the receive buffer of SPDR being empty, the RSPI copies the receive data in the shift register to the receive buffer and generates a receive buffer full interrupt request (SPRI). Since the shift register becomes empty upon completion of serial transfer, when the transmit buffer had been full before the serial transfer ended, the RSPI copies the data in the transmit buffer to the shift register. Even when received data is not copied from the shift register to the receive buffer in an overrun error status, upon completion of the serial transfer, the RSPI determines that the shift register is empty, thus data transfer from the transmit buffer to the shift register is enabled.
5. When SPDR is read by the receive buffer full interrupt routine, the receive data can be read.

If SPDR is written to when the transmit buffer holds data that has not yet been transmitted, the RSPI does not update the data in the transmit buffer. When writing to SPDR, make sure to use a transmit buffer empty interrupt request. To use a transmit buffer empty interrupt, set the SPTIE bit in SPCR to 1.
If the RSPI function is disabled (the SPCR.SPE bit being 0), set the SPTIE bit to 0.
When serial transfer ends with the receive buffer being full, the RSPI does not copy data from the shift register to the receive buffer, and detects an overrun error (refer to section 28.3.8, Error Detection). To prevent a receive data overrun error, read the received data using a receive buffer full interrupt request before the next serial transfer ends. To use an RSPI receive interrupt, set the SPCR.SPRIE bit to 1.
Transmission and reception interrupts or the corresponding IRn.IR flags (where n is the interrupt vector number) in the ICU can be used to confirm the states of the transmission and reception buffers. Refer to section 14, Interrupt Controller (ICUb), for the interrupt vector numbers.

### 28.3.8 Error Detection

In the normal RSPI serial transfer, the data written to the transmit buffer of SPDR is transmitted, and the received data can be read from the receive buffer of SPDR. If access is made to SPDR, depending on the status of the transmit/receive buffer or the status of the RSPI at the beginning or end of serial transfer, in some cases non-normal transfers can be executed.

If a non-normal transfer operation occurs, the RSPI detects the event as an overrun error, parity error, or mode fault error. Table 28.8 lists the relationship between non-normal transfer operations and the RSPI's error detection function.

**Table 28.8 Relationship between Non-Normal Transfer Operations and RSPI Error Detection Function**

| | Occurrence Condition | RSPI Operation | Error Detection |
|---|---|---|---|
| 1 | SPDR is written when the transmit buffer is full. | • The contents of the transmit buffer are kept.<br>• Missing write data. | None |
| 2 | Serial transfer is started in slave mode when transmit data is still not loaded on the shift register. | Data received in previous serial transfer is transmitted. | None |
| 3 | SPDR is read when the receive buffer is empty. | Previously received data is output. | None |
| 4 | Serial transfer terminates when the receive buffer is full. | • The contents of the receive buffer are kept.<br>• Missing receive data. | Overrun error |
| 5 | An incorrect parity bit is received when performing full-duplex synchronous serial communications with the parity function enabled. | The parity error flag is asserted. | Parity error |
| 6 | The SSLA0 input signal is asserted when the serial transfer is idle in multi-master mode. | • Driving of the RSPCKA, MOSIA, SSLA1 to SSLA3 output signals is stopped.<br>• RSPI function is disabled. | Mode fault error |
| 7 | The SSLA0 input signal is asserted during serial transfer in multi-master mode. | • Serial transfer is suspended.<br>• Missing transmit/receive data.<br>• Driving of the RSPCKA, MOSIA, SSLA1 to SSLA3 output signals is stopped.<br>• RSPI function is disabled. | Mode fault error |
| 8 | The SSLA0 input signal is negated during serial transfer in slave mode. | • Serial transfer is suspended.<br>• Missing transmit/receive data.<br>• Driving of the MISOA output signal is stopped.<br>• RSPI function is disabled. | Mode fault error |

On operation 1 described in Table 28.8, the RSPI does not detect an error. To prevent data omission during the writing to SPDR, write operations to SPDR should be executed using a transmit buffer empty interrupt request.

Likewise, the RSPI does not detect an error on operation 2. In a serial transfer that was started before the shift register was updated, the RSPI sends the data that was received in the previous serial transfer, and does not treat the operation indicated in 2 as an error. Note that the received data from the previous serial transfer is retained in the receive buffer of SPDR, thus it can be correctly read (if SPDR is not read before the end of the serial transfer, an overrun error may occur).

Similarly, the RSPI does not detect an error on operation 3. To prevent extraneous data from being read, SPDR read operation should be executed using a receive interrupt request.

An overrun error shown in 4 is described in section 28.3.8.1, Overrun Error. A parity error shown in 5 is described in section 28.3.8.2, Parity Error. A mode fault error shown in 6 to 8 is described in section 28.3.8.3, Mode Fault Error. For the transmit and receive interrupts, refer to section 28.3.7, Transmit Buffer Empty/Receive Buffer Full Interrupts.

### 28.3.8.1 Overrun Error

If a serial transfer ends when the receive buffer of SPDR is full, the RSPI detects an overrun error, and sets the OVRF flag in SPSR to 1. When the OVRF flag is 1, the RSPI does not copy data from the shift register to the receive buffer so that the data prior to the occurrence of the error is retained in the receive buffer. To set the OVRF flag to 0, write 0 to the OVRF flag after the CPU has read SPSR with the OVRF flag set to 1.

Figure 28.27 shows an example of operation of the OVRF flag. The SPSR and SPDR accesses shown in Figure 28.27 indicate the condition of accesses to SPSR and SPDR, respectively, where W denotes a write cycle, and R a read cycle. In the example in Figure 28.27, the RSPI performs an 8-bit serial transfer in which the SPCMDm.CPHA bit is 1 and the SPCMDm.CPOL bit is 0. The numbers given under the RSPCKA waveform represent the number of RSPCK cycles (i.e., the number of transferred bits).

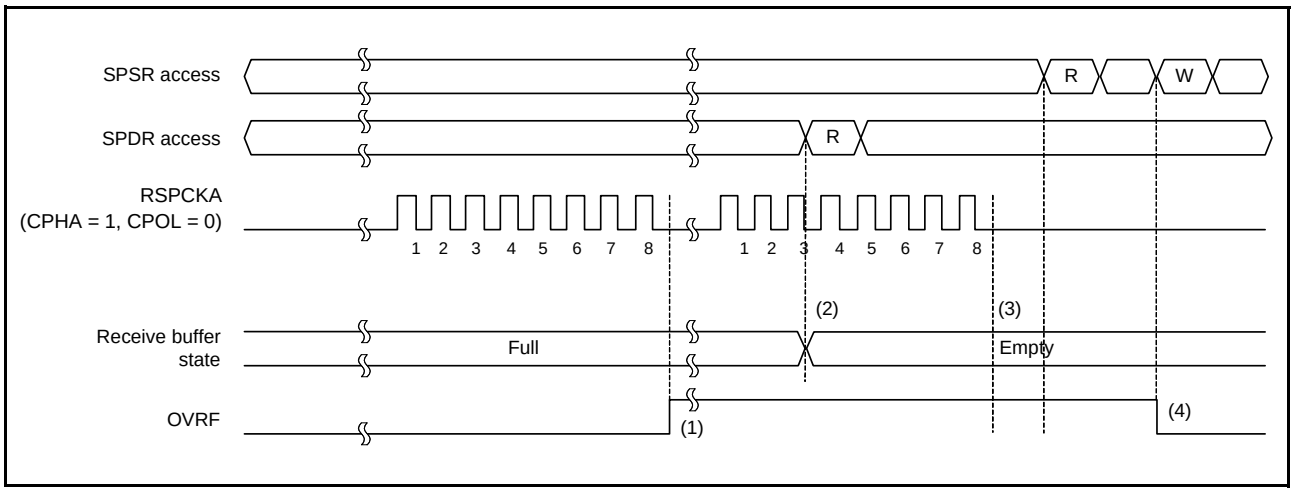



**Figure 28.27 Operation Example of OVRF Flag**

The operation of the flags at the timing shown in steps (1) to (4) in the figure is described below.

1. If a serial transfer terminates with the receive buffer full, the RSPI detects an overrun error, and sets the OVRF flag to 1. The RSPI does not copy the data in the shift register to the receive buffer. Even if the SPPE bit is 1, parity errors are not detected. In master mode, the RSPI copies the pointer value to SPCMDm register to the SPSSR.SPECM[2:0] bits.
2. When SPDR is read, the RSPI outputs the data in the receive buffer can be read. The receive buffer becoming empty does not clear the OVRF flag.
3. If the serial transfer ends with the OVRF flag being 1 (an overrun error occurs), the RSPI does not copy the data in the shift register to the receive buffer. A reception-buffer interrupt is not generated. Even if the SPPE bit is 1, parity errors are not detected. When in master mode, the RSPI does not update the SPSSR.SPECM[2:0] bits. When in an overrun error state and the RSPI does not copy the received data from the shift register to the receive buffer, upon termination of the serial transfer, the RSPI determines that the shift register is empty; in this manner, data transfer from the transmit buffer to the shift register is enabled.
4. If the value 0 is written to the OVRF flag after SPSR is read when the OVRF flag is 1, the OVRF flag is cleared to 0.

The occurrence of an overrun can be checked either by reading SPSR or by using an RSPI error interrupt and reading SPSR. When executing a serial transfer, measures should be taken to ensure the early detection of overrun errors, such as reading SPSR immediately after SPDR is read. When the RSPI is used in master mode, the pointer value to SPCMDm register at the occurrence of the error can be checked by reading the SPSSR.SPECM[2:0] bits.

If an overrun error occurs and the OVRF flag is set to 1, normal reception operations cannot be performed until the OVRF flag is cleared.

### 28.3.8.2 Parity Error

If full-duplex synchronous serial communications is performed with the SPCR.TXMD bit cleared to 0 and the SPCR2.SPPE bit set to 1, when serial transfer ends, the RSPI checks whether there are parity errors. Upon detecting a parity error in the received data, the RSPI sets the SPSR.PERF flag to 1. Since the RSPI does not copy the data in the shift register to the receive buffer when the SPSR.OVRF flag is set to 1, parity error detection is not performed for the received data. To set the PERF flag to 0, write 0 to the PERF flag after SPSR register is read with the PERF flag set to 1. Figure 28.28 shows an example of operation of the OVRF and PERF flags. The SPSR access shown in Figure 28.28 indicates the condition of access to SPSR register, where W denotes a write cycle, and R a read cycle. In the example of Figure 28.28, full-duplex synchronous serial communications is performed while the SPCR.TXMD bit is 0 and the SPCR2.SPPE bit is 1. The RSPI performs an 8-bit serial transfer in which the SPCMDm.CPHA bit is 1 and the SPCMDm.CPOL bit is 0. The numbers given under the RSPCKA waveform represent the number of RSPCK cycles (i.e., the number of transferred bits).

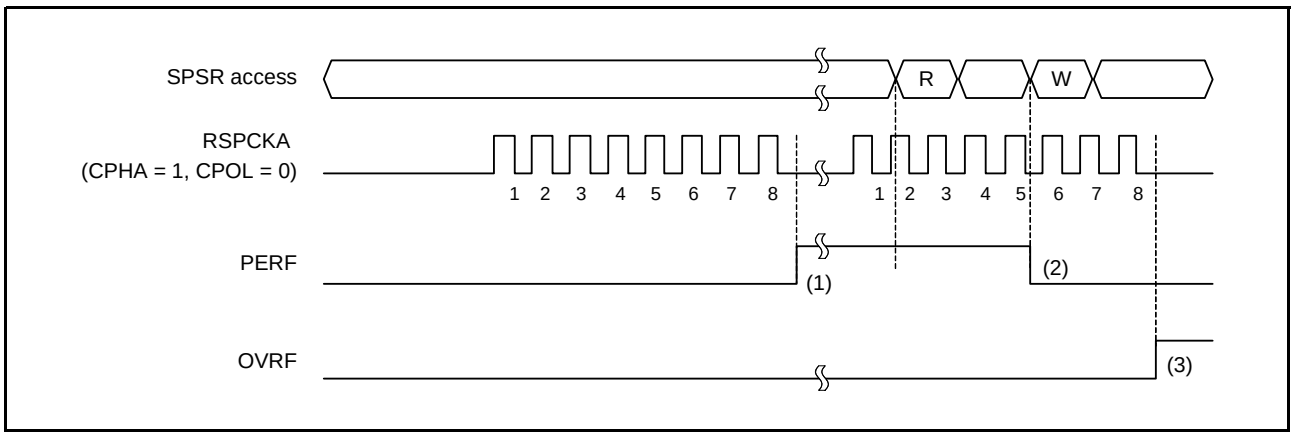



**Figure 28.28 Operation Example of PERF Flag**

The operation of the flags at the timing shown in steps (1) to (3) in the figure is described below.

1. If a serial transfer terminates with the RSPI not detecting an overrun error, the RSPI copies the data in the shift register to the receive buffer. The RSPI judges the received data at this timing, and sets the PERF flag to 1 if a parity error is detected. In master mode, the RSPI copies the pointer value to SPCMDm register to the SPSSR.SPECM[2:0] bits.
2. If the value 0 is written to the PERF flag after SPSR register is read when the PERF flag is 1, the PERF flag is cleared to 0.
3. When the RSPI detects an overrun error and serial transfer is terminated, the data in the shift register is not copied to the receive buffer. The RSPI does not perform parity error detection at this timing.

The occurrence of a parity error can be checked either by reading SPSR register or by using an RSPI error interrupt and reading SPSR register. When executing a serial transfer, measures should be taken to ensure the early detection of parity errors, such as reading SPSR. When the RSPI is used in master mode, the pointer value to SPCMDm register at the occurrence of the error can be checked by reading the SPSSR.SPECM[2:0] bits.

#### 28.3.8.3 Mode Fault Error

The RSPI operates in multi-master mode when the SPCR.MSTR bit is 1, the SPCR.SPMS bit is 0, and the SPCR.MODFEN bit is 1. If the active level is input with respect to the SSLA0 input signal of the RSPI in multi-master mode, the RSPI detects a mode fault error irrespective of the status of the serial transfer, and sets the MODF bit in the RSPI status register (SPSR) to 1. Upon detecting the mode fault error, the RSPI copies the value of the pointer to SPCMDm to the SPSSR.SPECM[2:0] bits. The active level of the SSLA0 signal is determined by the SSL0P bit in the RSPI slave select polarity register (SSLP).

When the MSTR bit is 0, the RSPI operates in slave mode. The RSPI detects a mode fault error if the MODFEN bit in the RSPI in slave mode is 1, and the SPMS bit is 0, and if the SSL0 input signal is negated during the serial transfer period (from the time the driving of valid data is started to the time the final valid data is fetched).

Upon detecting a mode fault error, the RSPI stops driving of the output signals and clears the SPCR.SPE bit to 0 (refer to section 28.3.9, Initializing RSPI). In the case of multi-master configuration, detection of a mode fault error is used to stop driving of the output signals and the RSPI function, which allows the master right to be released.

The occurrence of a mode fault error can be checked either by reading SPSR or by using an RSPI error interrupt and reading SPSR. Detecting mode-fault errors without utilizing the RSPI error interrupt requires polling of SPSR. When using the RSPI in master mode, the pointer value to SPCMDm register at the occurrence of the error can be checked by reading the SPSSR.SPECM[2:0] bits.

When the MODF bit is 1, writing of the value 1 to the SPE bit is ignored by the RSPI. To enable the RSPI function after the detection of a mode fault error, the MODF bit must be set to 0.

### 28.3.9 Initializing RSPI

If the value 0 is written to the SPCR.SPE bit or the RSPI clears the SPE bit to 0 because of the detection of a mode fault error, the RSPI disables the RSPI function, and initializes some of the module functions. When a system reset is generated, the RSPI initializes all of the module functions. The following describes initialization by the clearing of the SPCR.SPE bit and initialization by a system reset.

#### 28.3.9.1 Initialization by Clearing the SPE Bit

When the SPCR.SPE bit is cleared, the RSPI performs the following initialization:

- Suspending any serial transfer that is being executed
- Stopping the driving of output signals (Hi-Z) in slave mode
- Initializing the internal state of the RSPI
- Initializing the transmit buffer of the RSPI

Initialization by the clearing of the SPE bit does not initialize the control bits of the RSPI. For this reason, the RSPI can be started in the same transfer mode as prior to the initialization if the SPE bit is set to 1 again.
The SPSR.OVRF and SPSR.MODF flags are not initialized, nor is the value of the RSPI sequence status register (SPSSR) initialized. For this reason, even after the RSPI is initialized, data from the receive buffer can be read in order to check the status of error occurrence during an RSPI transfer.
The transmit buffer is initialized to an empty state. Therefore, if the SPCR.SPTIE bit is set to 1 after RSPI initialization, a transmit buffer empty interrupt is generated. When the RSPI is initialized, in order to disable any transmit buffer empty interrupt, the value 0 should be written to the SPTIE bit simultaneously with the writing of the value 0 to the SPE bit. To disable any transmit buffer empty interrupt after a mode fault error is detected, use an error handling routine to write the value 0 to the SPTIE bit.

#### 28.3.9.2 System Reset

The initialization by a system reset completely initializes the RSPI through the initialization of all bits for controlling the RSPI, initialization of the status bits, and initialization of data registers, in addition to the requirements described in section 28.3.9.1, Initialization by Clearing the SPE Bit.

## 28.3.10 SPI Operation

### 28.3.10.1 Master Mode Operation

The only difference between single-master mode operation and multi-master mode operation lies in mode fault error detection (refer to section 28.3.8, Error Detection). When operating in single-master mode, the RSPI does not detect mode fault errors whereas the RSPI running in multi-master mode does detect mode fault errors. This section explains operations that are common to single-master mode and multi-master mode.

#### (1) Starting a Serial Transfer

The RSPI updates the data in the transmit buffer (SPTX) when data is written to the RSPI data register (SPDR) with the RSPI transmit buffer being empty (data for the next transfer is not set). When the shift register is empty after the number of frames set in the SPDCR.SPFC[1:0] bits are written to the SPDR, the RSPI copies data from the transmission buffer to the shift register and starts serial transfer. Upon copying transmit data to the shift register, the RSPI changes the status of the shift register to “full”, and upon termination of serial transfer, it changes the status of the shift register to “empty”. The status of the shift register cannot be referenced.
For details on the RSPI transfer format, refer to section 28.3.5, Transfer Format. The polarity of the SSLAi output pins depends on the SSLP register settings.

#### (2) Terminating a Serial Transfer

Irrespective of the SPCMDm.CPHA bit the RSPI terminates the serial transfer after transmitting an RSPCKA edge corresponding to the final sampling timing. If free space is available in the receive buffer (SPRX), upon termination of serial transfer, the RSPI copies data from the shift register to the receive buffer of the RSPI data register (SPDR).
It should be noted that the final sampling timing varies depending on the bit length of transfer data. In master mode, the RSPI data length depends on the SPCMDm.SPB[3:0] bit setting. The polarity of the SSLAi output pin depends on the SSLP register settings.
For details on the RSPI transfer format, refer to section 28.3.5, Transfer Format.

### (3) Sequence Control

The transfer format that is employed in master mode is determined by SPSCR, SPCMDm, SPBR, SPCKD, SSLND, and SPND registers.

SPSCR is a register used to determine the sequence configuration for serial transfers that are executed by the RSPI in master mode. The following items are set in SPCMDm register: SSLAi pin output signal value, MSB/LSB first, data length, some of the bit rate settings, RSPCK polarity/phase, whether SPCKD is to be referenced, whether SSLND is to be referenced, and whether SPND is to be referenced. SPBR holds some of the bit rate settings; SPCKD, an RSPI clock delay value; SSLND, an SSL negation delay; and SPND, a next-access delay value.

According to the sequence length that is assigned to SPSCR, the RSPI makes up a sequence comprised of a part or all of SPCMDm register. The RSPI contains a pointer to the SPCMDm register that makes up the sequence. The value of this pointer can be checked by reading the SPSSR.SPCP[2:0] bits. When the SPCR.SPE bit is set to 1 and the RSPI function is enabled, the RSPI loads the pointer to the commands in SPCMD0, and incorporates the SPCMD0 settings into the transfer format at the beginning of serial transfer. The RSPI increments the pointer each time the next-access delay period for a data transfer ends. Upon completion of the serial transfer that corresponds to the final command comprising the sequence, the RSPI sets the pointer in SPCMD0, and in this manner the sequence is executed repeatedly.

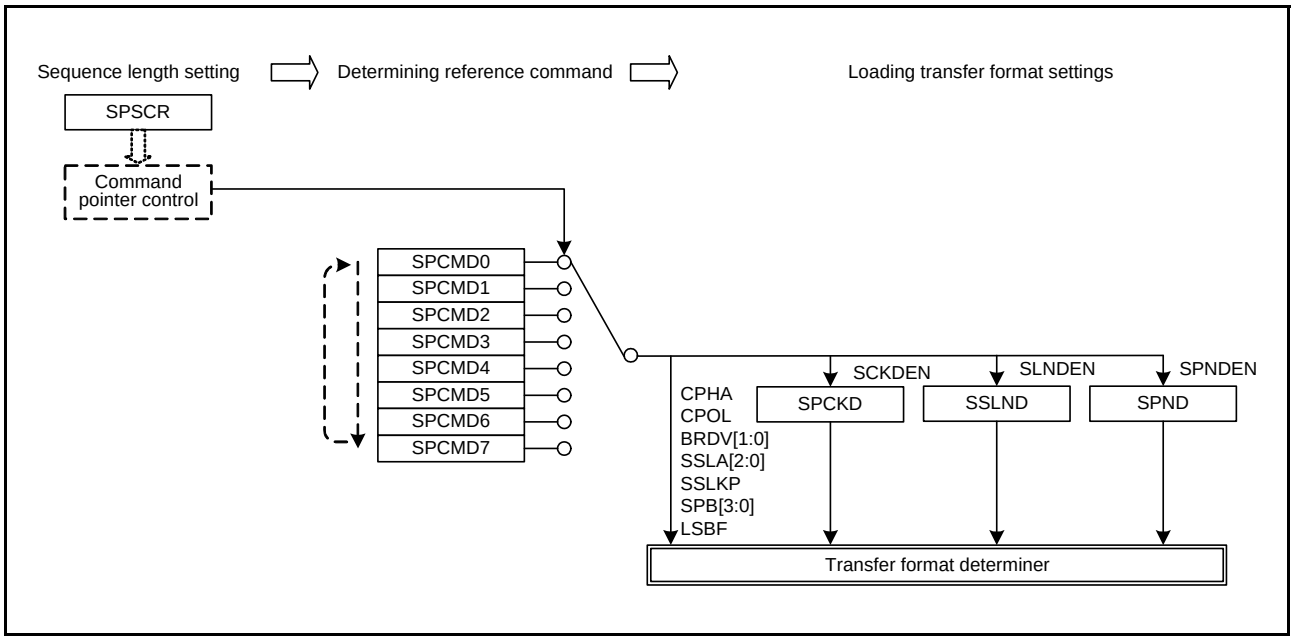



**Figure 28.29 Procedure for Determining the Form of Serial Transfer in Master Mode**

In this section, a frame is the combination of the data (SPDR) and the settings (SPCMDm).

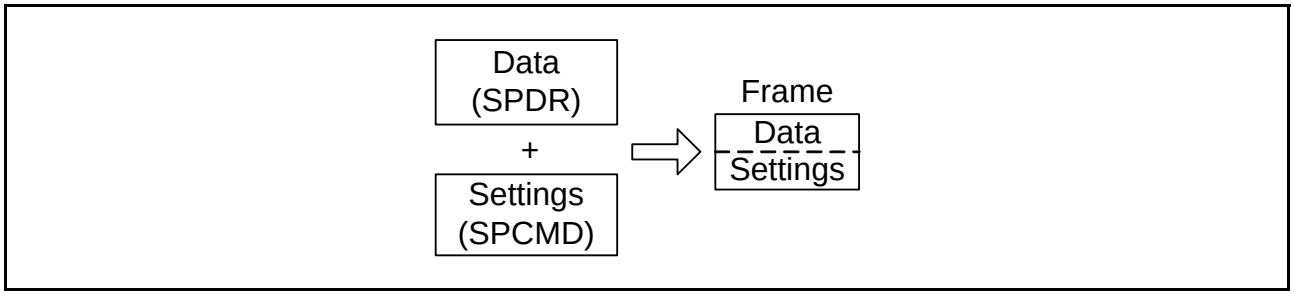



**Figure 28.30 Concept of a Frame**

Figure 28.31 shows the relationship between the command and the transmit and receive buffers in the sequence of operations specified by the settings in Table 28.4.

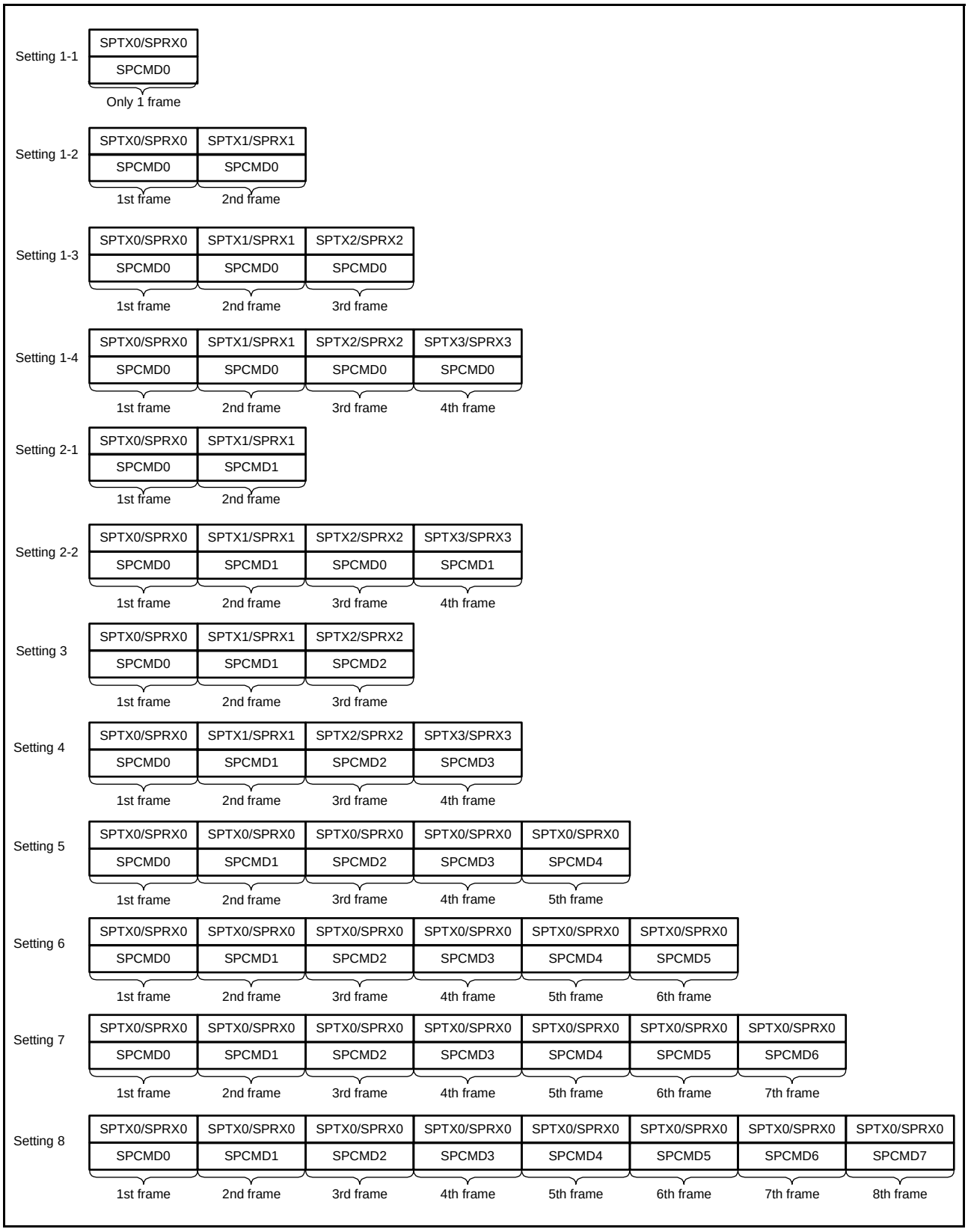



**Figure 28.31 Correspondence between the RSPI Command Register and Transmit/Receive Buffers in Sequence Operations**

### (4) Burst Transfer

If the SPCMDm.SSLKP bit that the RSPI references during the current serial transfer is 1, the RSPI keeps the SSLAi signal level during the serial transfer until the beginning of the SSLAi signal assertion for the next serial transfer. If the SSLAi signal level for the next serial transfer is the same as the SSLAi signal level for the current serial transfer, the RSPI can execute continuous serial transfers while keeping the SSLAi signal assertion status (burst transfer).
Figure 28.32 shows an example of an SSLAi signal operation for the case where a burst transfer is implemented using SPCMD0 and SPCMD1 register settings. The text below explains the RSPI operations (1) to (7) as shown in Figure 28.32. It should be noted that the polarity of the SSLAi output signal depends on the SSLP register settings.

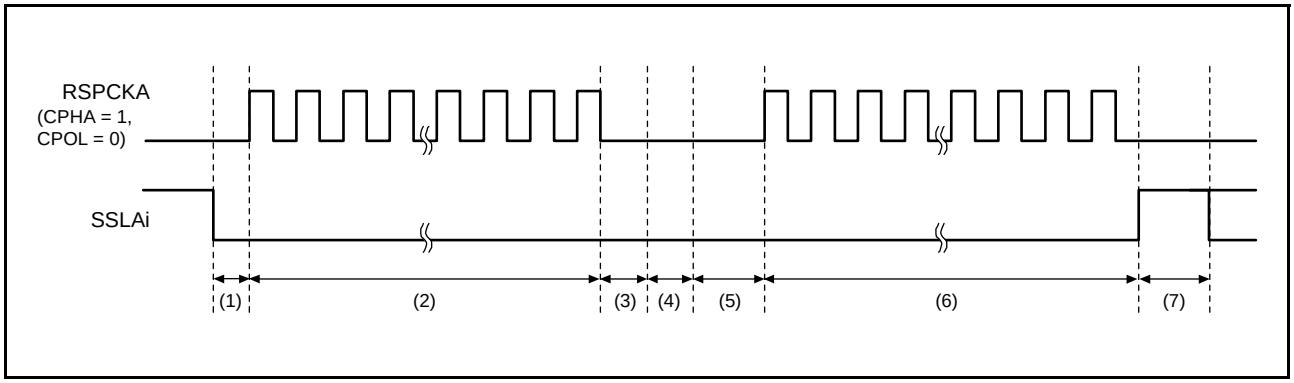



**Figure 28.32 Example of Burst Transfer Operation Using SSLKP Bit**

(1) Based on SPCMD0, the RSPI asserts the SSLAi signal and inserts RSPCK delays.
(2) The RSPI executes serial transfers according to SPCMD0.
(3) The RSPI inserts SSL negation delays.
(4) Since the SPCMD0.SSLKP bit is 1, the RSPI keeps the SSLAi signal value on SPCMD0. This period is sustained, at the shortest, for a period equal to the next-access delay of SPCMD0. If the shift register is empty after the passage of a minimum period, this period is sustained until the transmit data is stored in the shift register for the next transfer.
(5) Based on SPCMD1, the RSPI asserts the SSLAi signal and inserts RSPCK delays.
(6) The RSPI executes serial transfers according to SPCMD1.
(7) Because the SPCMD1.SSLKP bit is 0, the RSPI negates the SSLAi signal. In addition, a next-access delay is inserted according to SPCMD1.

If the SSLAi signal output settings in the SPCMDm register in which 1 is assigned to the SSLKP bit are different from the SSLAi signal output settings in the SPCMDm register to be used in the next transfer, the RSPI switches the SSLAi signal status to SSLAi signal assertion ((5) in Figure 28.32) corresponding to the command for the next transfer. Note that if such an SSLAi signal switching occurs, the slaves that drive the MISOA signal compete, and collision of signal levels may occur.
The RSPI in master mode references the SSLAi signal operation within the module for the case where the SSLKP bit is not used. Even when the SPCMDm.CPHA bit is 0, the RSPI can accurately start serial transfers by using the SSLAi signal assertion for the next transfer that is detected internally.

### (5) RSPCK Delay (t1)

The RSPCK delay value of the RSPI in master mode depends on the SPCMDm.SCKDEN bit setting and the SPCKD register setting. The RSPI determines the SPCMDm register to be referenced during serial transfer by pointer control, and determines an RSPCK delay value during serial transfer by using the SPCMDm.SCKDEN bit and SPCKD, as listed in Table 28.9. For a definition of RSPCK delay, refer to section 28.3.5, Transfer Format.

**Table 28.9 Relationship among SCKDEN Bit, SPCKD, and RSPCK Delay Value**

| SPCMDm.SCKDEN Bit | SPCKD.SCKDL[2:0] Bits | RSPCK Delay Value |
|---|---|---|
| 0 | 000 to 111 | 1 RSPCK |
| 1 | 000 | 1 RSPCK |
| | 001 | 2 RSPCK |
| | 010 | 3 RSPCK |
| | 011 | 4 RSPCK |
| | 100 | 5 RSPCK |
| | 101 | 6 RSPCK |
| | 110 | 7 RSPCK |
| | 111 | 8 RSPCK |

### (6) SSL Negation Delay (t2)

The SSL negation delay value of the RSPI in master mode depends on the SPCMDm.SLNDEN bit setting and the SSLND register setting. The RSPI determines the SPCMDm register to be referenced during serial transfer by pointer control, and determines an SSL negation delay value during serial transfer by using the SPCMDm.SLNDEN bit and SSLND, as listed in Table 28.10. For a definition of SSL negation delay, refer to section 28.3.5, Transfer Format.

**Table 28.10 Relationship among SLNDEN Bit, SSLND, and SSL Negation Delay Value**

| SPCMDm.SLNDEN Bit | SSLND.SLNDL[2:0] Bits | SSL Negation Delay Value |
|---|---|---|
| 0 | 000 to 111 | 1 RSPCK |
| 1 | 000 | 1 RSPCK |
| | 001 | 2 RSPCK |
| | 010 | 3 RSPCK |
| | 011 | 4 RSPCK |
| | 100 | 5 RSPCK |
| | 101 | 6 RSPCK |
| | 110 | 7 RSPCK |
| | 111 | 8 RSPCK |

### (7) Next-Access Delay (t3)

The next-access delay value of the RSPI in master mode depends on the SPCMDm.SPNDEN bit setting and the SPND setting. The RSPI determines the SPCMDm register to be referenced during serial transfer by pointer control, and determines a next-access delay value during serial transfer by using the SPCMDm.SPNDEN bit and SPND, as listed in Table 28.11. For a definition of next-access delay, refer to section 28.3.5, Transfer Format.

**Table 28.11 Relationship among SPNDEN Bit, SPND, and Next-Access Delay Value**

| SPCMDm.SPNDEN Bit | SPND.SPNDL[2:0] Bits | Next-Access Delay Value |
|---|---|---|
| 0 | 000 to 111 | 1 RSPCK + 2 PCLK |
| 1 | 000 | 1 RSPCK + 2 PCLK |
| | 001 | 2 RSPCK + 2 PCLK |
| | 010 | 3 RSPCK + 2 PCLK |
| | 011 | 4 RSPCK + 2 PCLK |
| | 100 | 5 RSPCK + 2 PCLK |
| | 101 | 6 RSPCK + 2 PCLK |
| | 110 | 7 RSPCK + 2 PCLK |
| | 111 | 8 RSPCK + 2 PCLK |

### (8) Initialization Flowchart

Figure 28.33 is a flowchart illustrating an example of initialization in SPI operation when the RSPI is used in master mode. For a description of how to set up the interrupt controller and I/O ports, refer to the descriptions given in the individual blocks.

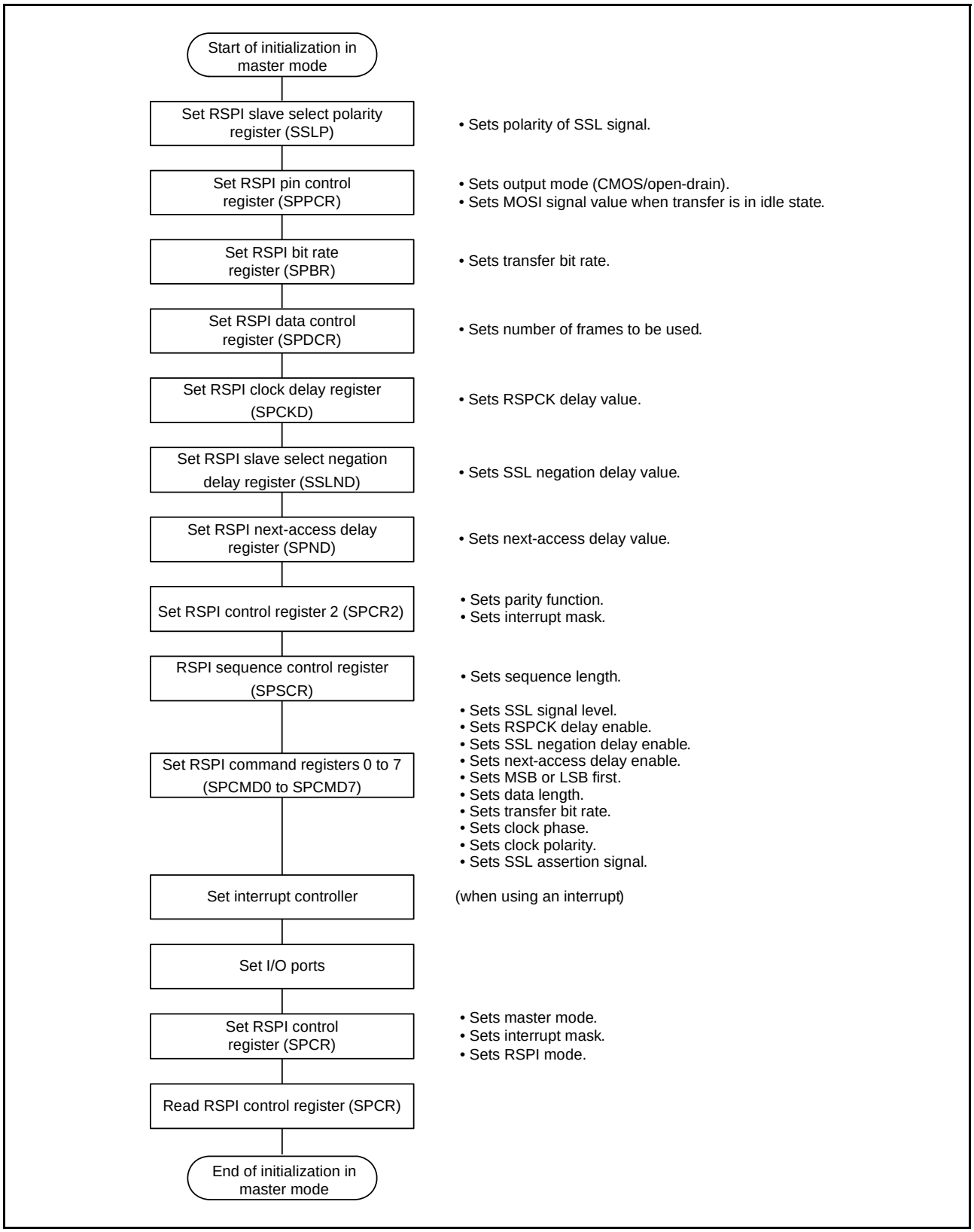



**Figure 28.33 Example of Initialization Flowchart in Master Mode (SPI Operation)**

### (9) Software Processing Flow

Figure 28.34 to Figure 28.36 show examples of the flow of software processing.

#### (a) Transmit Processing Flow

When transmitting data, the CPU will be notified of the completion of data transmission after the last writing of data for transmission if the SPII interrupt is enabled.

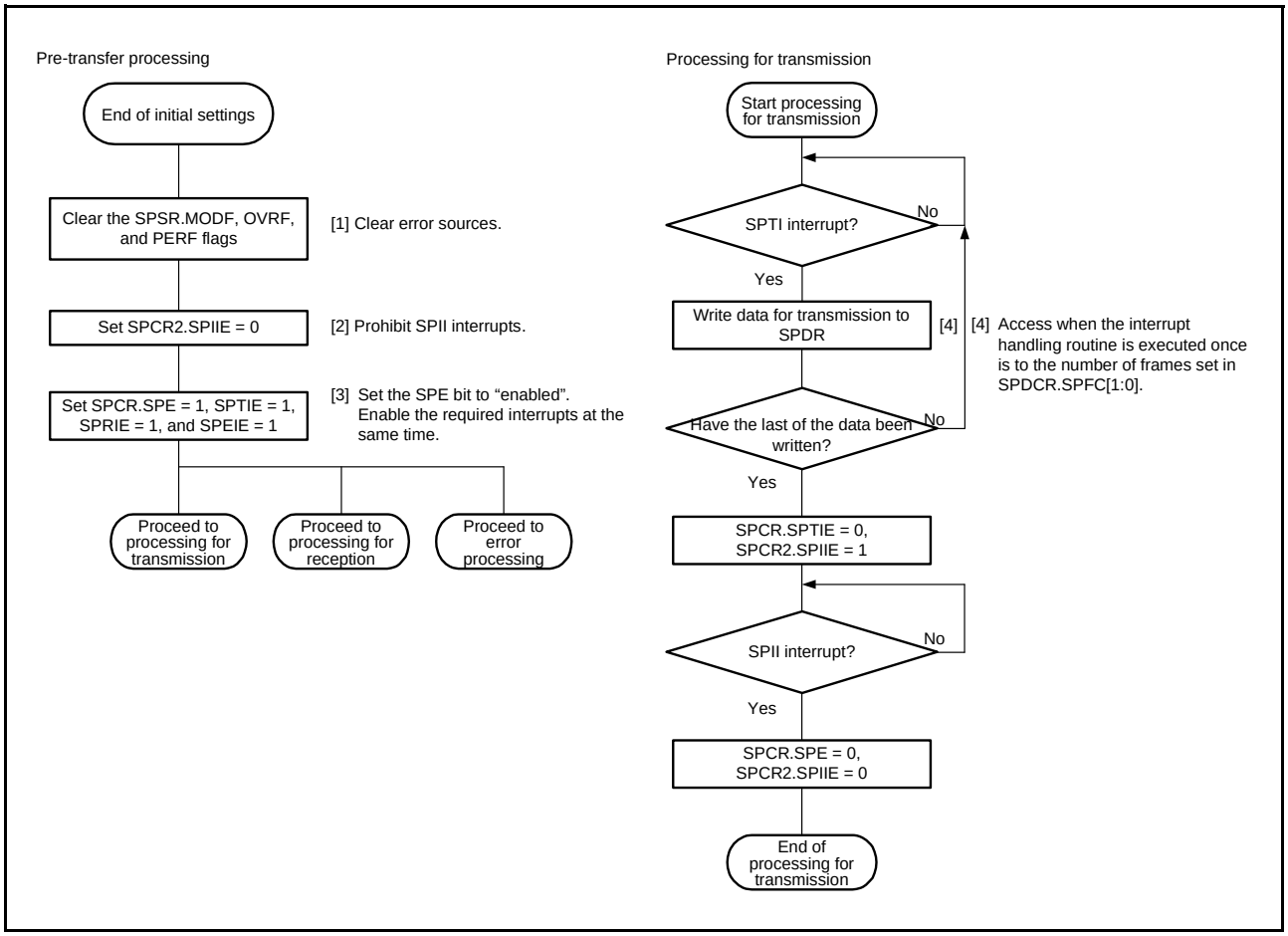



**Figure 28.34 Flowchart in Master Mode (Transmission)**

## (b) Receive Processing Flow

The RSPI does not handle receive-only operation, so processing for transmission is required if reception is to proceed.

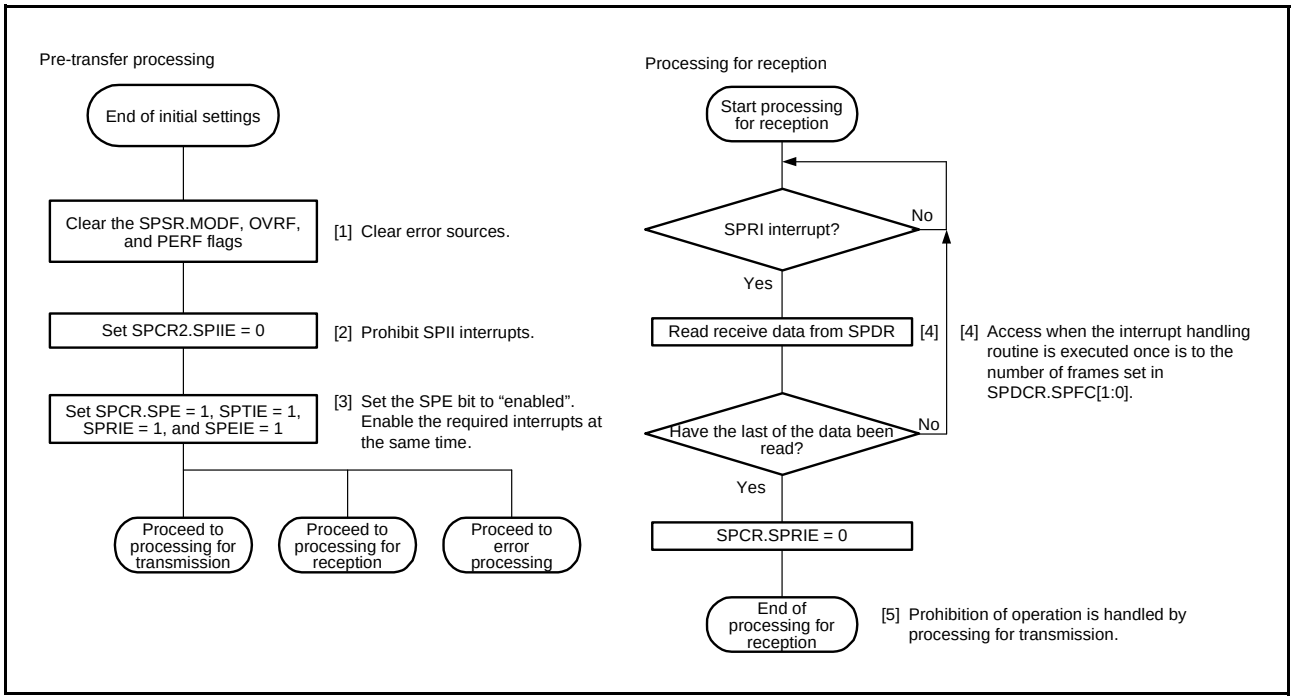



**Figure 28.35 Flowchart in Master Mode (Reception)**

### (c) Flow of error processing

The RSPI has three types of error. When a mode-fault error is generated, the SPCR.SPE bit is automatically cleared, stopping operations for transmission and reception. For errors from other sources, however, the SPCR.SPE bit is not cleared and operations for transmission and reception continue; accordingly, we recommend clearing of the SPCR.SPE bit to stop operations in the case of errors other than mode-fault errors. Not doing so will lead to updating of the SPSSR.SPECM[2:0] bits.

When an error occurs, clear the ICU.IRn.IR flag in the error processing routine. If this is not done, the ICU.IRn.IR flag may continue to indicate the SPTI interrupt or SPRI interrupt request. If the SPRI interrupt request is indicated, read the receive buffer and initialize the sequencer in the RSPI.

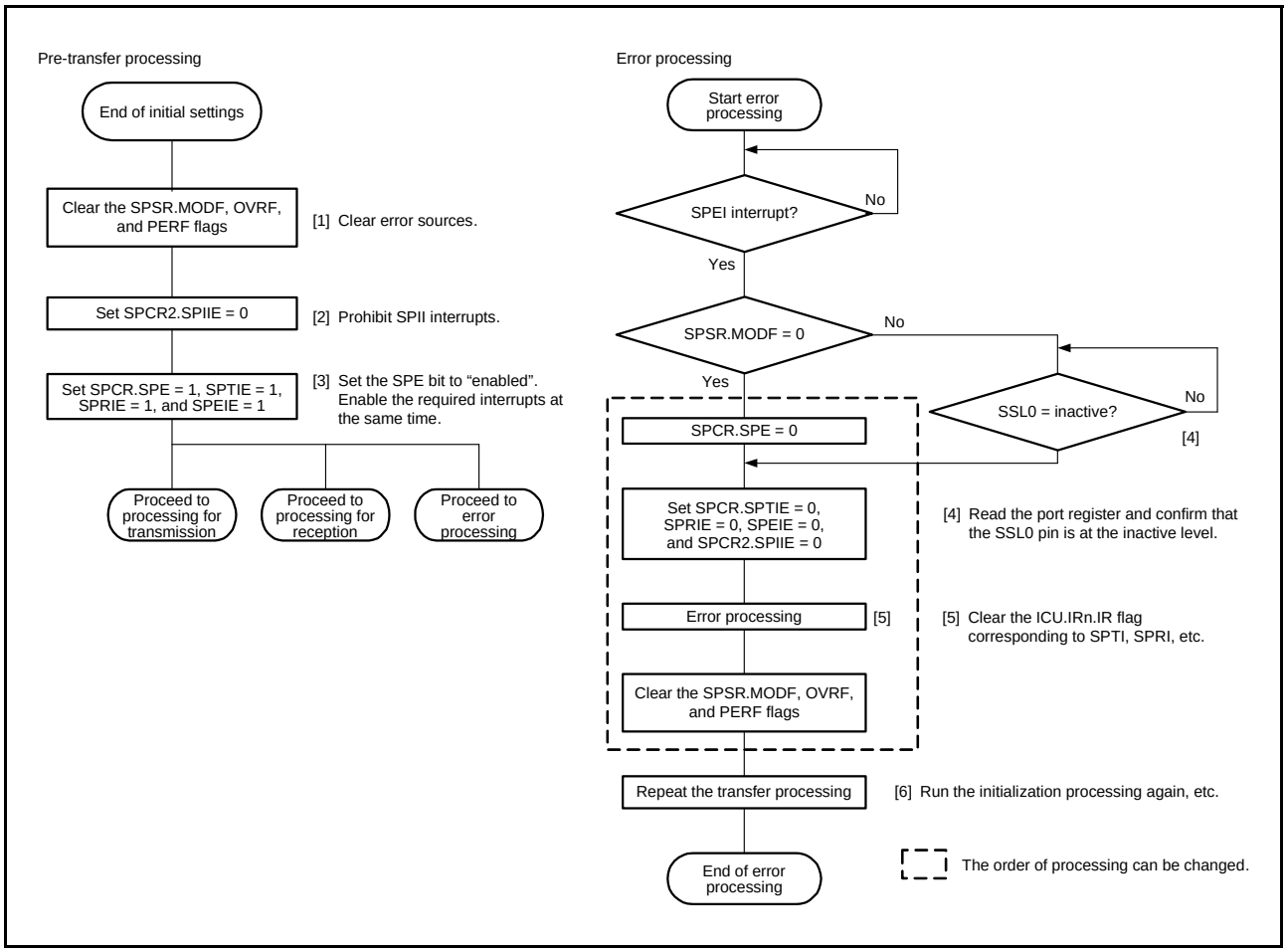



**Figure 28.36 Flowchart for Master Mode (Error Processing)**

### 28.3.10.2 Slave Mode Operation

#### (1) Starting a Serial Transfer

If the SPCMD0.CPHA bit is 0, when detecting an SSLA0 input signal assertion, the RSPI needs to start driving valid data to the MISOA output signal. For this reason, when the CPHA bit is 0, the assertion of the SSLA0 input signal triggers the start of a serial transfer.
If the CPHA bit is 1, when detecting the first RSPCKA edge in an SSLA0 signal asserted condition, the RSPI needs to start driving valid data to the MISOA output signal. For this reason, when the CPHA bit is 1, the first RSPCKA edge in an SSLA0 signal asserted condition triggers the start of a serial transfer.
When detecting the start of a serial transfer in a condition in which the shift register is empty, the RSPI changes the status of the shift register to “full”, so that data cannot be copied from the transmit buffer to the shift register when serial transfer is in progress. If the shift register was full before the serial transfer started, the RSPI leaves the status of the shift register unchanged, in the full state.
Irrespective of CPHA bit setting, the timing at which the RSPI starts driving of the MISOA output signal is the SSLA0 signal assertion timing. The data which is output by the RSPI is either valid or invalid, depending on the CPHA bit setting.
For details on the RSPI transfer format, refer to section 28.3.5, Transfer Format. The polarity of the SSLA0 input signal depends on the setting of the SSLP.SSL0P bit.

#### (2) Terminating a Serial Transfer

Irrespective of the SPCMD0.CPHA bit, the RSPI terminates the serial transfer after detecting an RSPCKA edge corresponding to the final sampling timing. When free space is available in the receive buffer, upon termination of serial transfer the RSPI copies received data from the shift register to the receive buffer of the RSPI data register (SPDR).
Upon termination of a serial transfer the RSPI changes the status of the shift register to “empty”, regardless of the receive buffer state. A mode fault error occurs if the RSPI detects an SSLA0 input signal negation from the beginning of serial transfer to the end of serial transfer (refer to section 28.3.8, Error Detection).
The final sampling timing changes depending on the bit length of transfer data. In slave mode, the RSPI data length depends on the SPCMD0.SPB[3:0] bit setting. The polarity of the SSLA0 input signal depends on the SSLP.SSL0P bit setting.
For details on the RSPI transfer format, refer to section 28.3.5, Transfer Format.

#### (3) Notes on Single-Slave Operations

If the SPCMD0.CPHA bit is 0, the RSPI starts serial transfers when it detects the assertion edge for an SSLA0 input signal. In the type of configuration shown in Figure 28.7 as an example, if the RSPI is used in single-slave mode, the SSLA0 signal is always fixed at the active state. Therefore, when the CPHA bit is set to 0, the RSPI cannot correctly start a serial transfer. To correctly execute transmit/receive operations by the RSPI in slave mode in a configuration in which the SSLA0 input signal is fixed at the active state, the CPHA bit should be set to 1. If there is a need for setting the CPHA bit to 0, the SSLA0 input signal should not be fixed.

### (4) Burst Transfer

If the SPCMD0.CPHA bit is 1, continuous serial transfer (burst transfer) can be executed while retaining the assertion state for the SSLA0 input signal. If the CPHA bit is 1, the period from the first RSPCKA edge to the sampling timing for the reception of the final bit in an SSLA0 signal active state corresponds to a serial transfer period. Even when the SSLA0 input signal remains at the active level, the RSPI can accommodate burst transfers because it can detect the start of an access.
If the CPHA bit is 0, the second and subsequent serial transfers during burst transfer cannot be executed correctly.

### (5) Initialization Flowchart

Figure 28.37 is a flowchart illustrating an example of initialization in SPI operation when the RSPI is used in slave mode. For a description of how to set up the interrupt controller and I/O ports, refer to the descriptions given in the individual blocks.

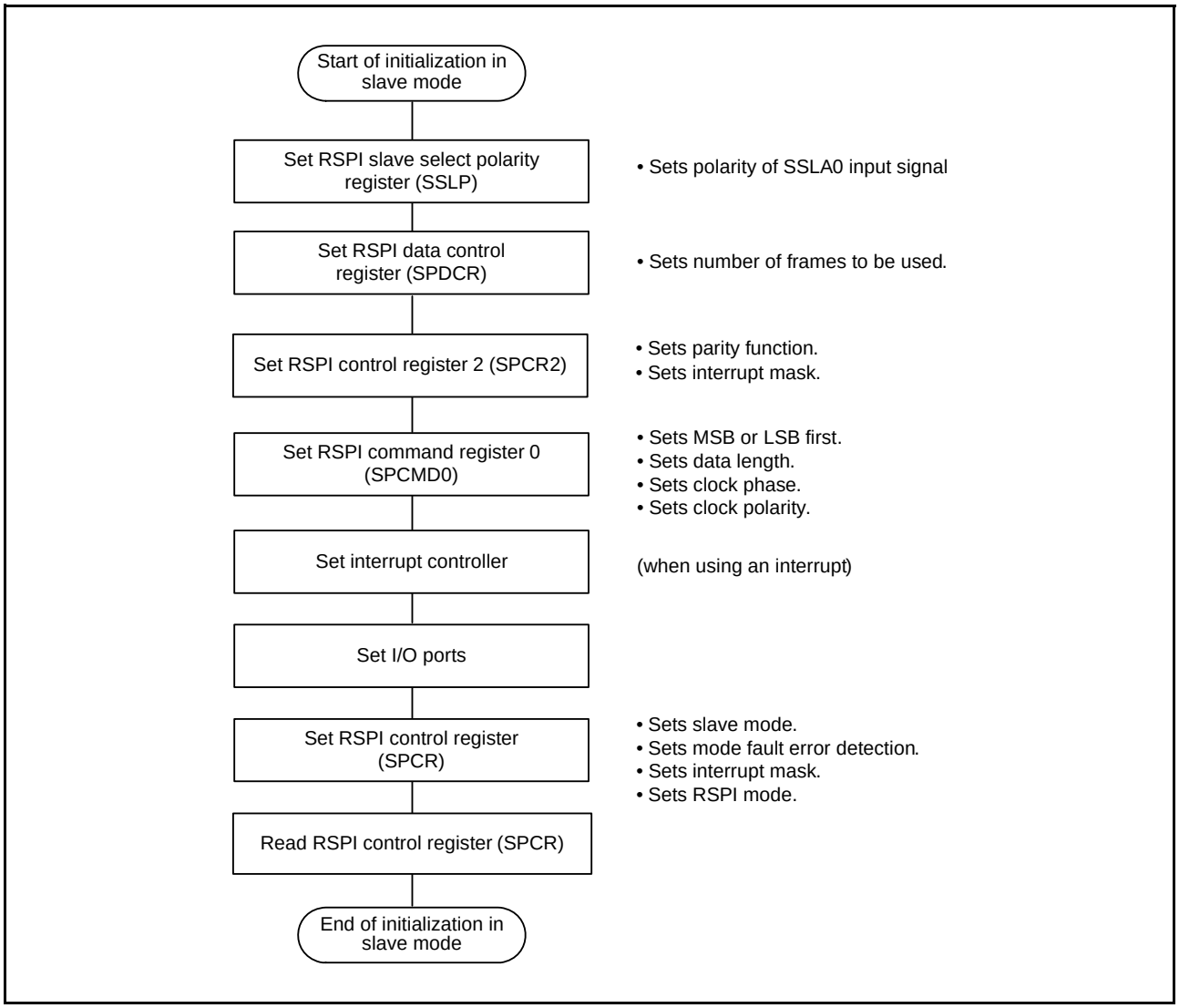



**Figure 28.37 Example of Initialization Flowchart in Slave Mode (SPI Operation)**

### (6) Software Processing Flow

Figure 28.38 to Figure 28.40 show examples of the flow of software processing.

#### (a) Transmit Processing Flow

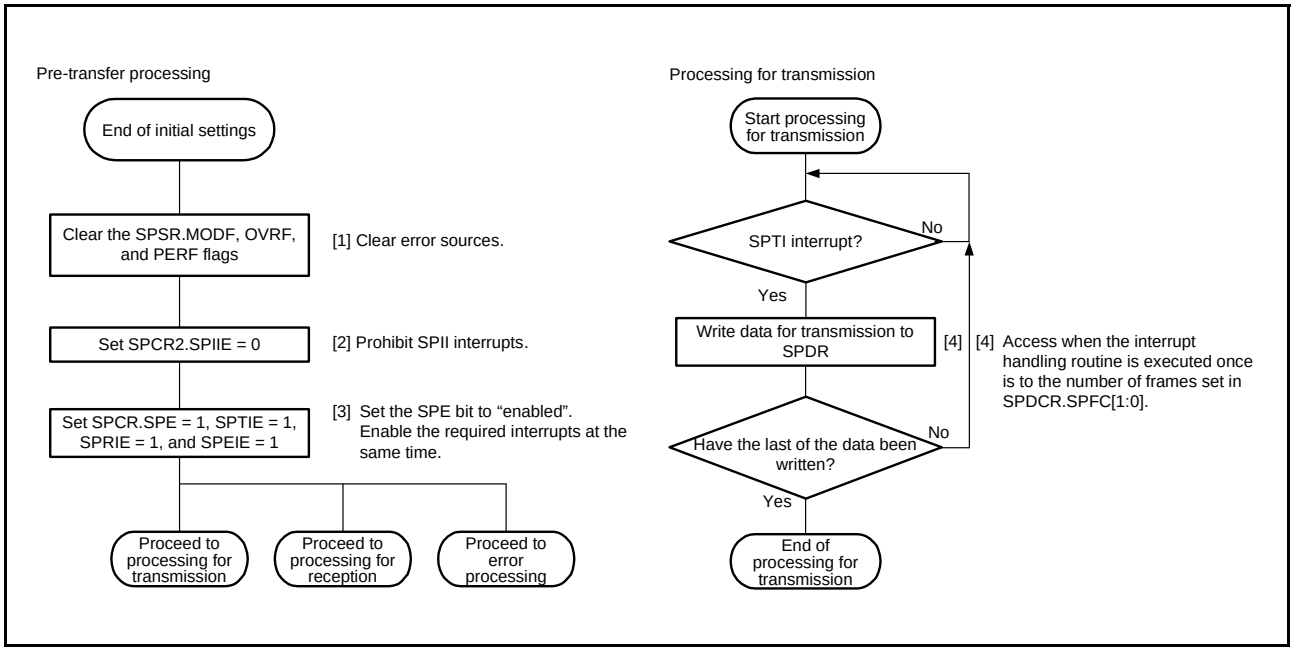



**Figure 28.38 Flowchart in Slave Mode (Transmission)**

#### (b) Receive Processing Flow

The RSPI does not handle receive-only operation, so processing for transmission is required if reception is to proceed.

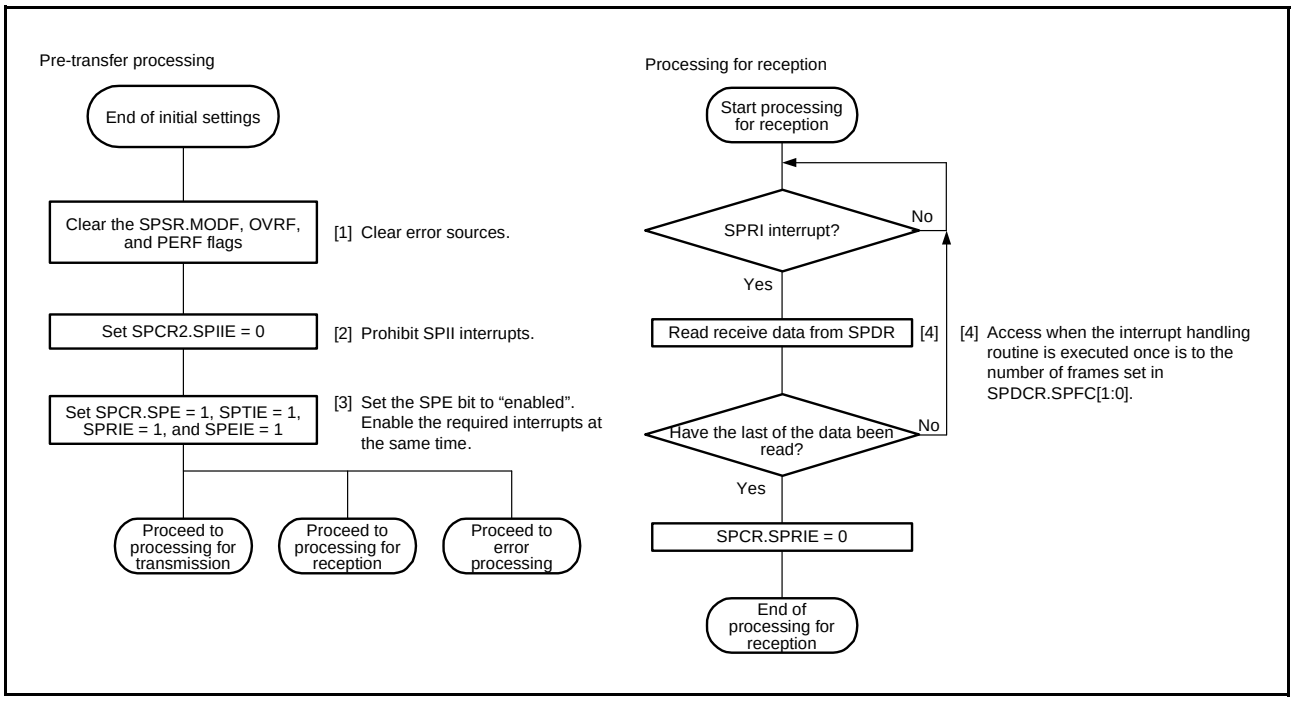



**Figure 28.39 Flowchart in Slave Mode (Reception)**

### (c) Flow of error processing

In slave operation, even when a mode-fault error is generated, the SPSR.MODF flag can be cleared without de-asserting the pin.

When an error occurs, clear the ICU.IRn.IR flag in the error processing routine. If this is not done, the ICU.IRn.IR flag may continue to indicate the SPTI interrupt or SPRI interrupt request. If the SPRI interrupt request is indicated, read the receive buffer and initialize the sequencer in the RSPI.

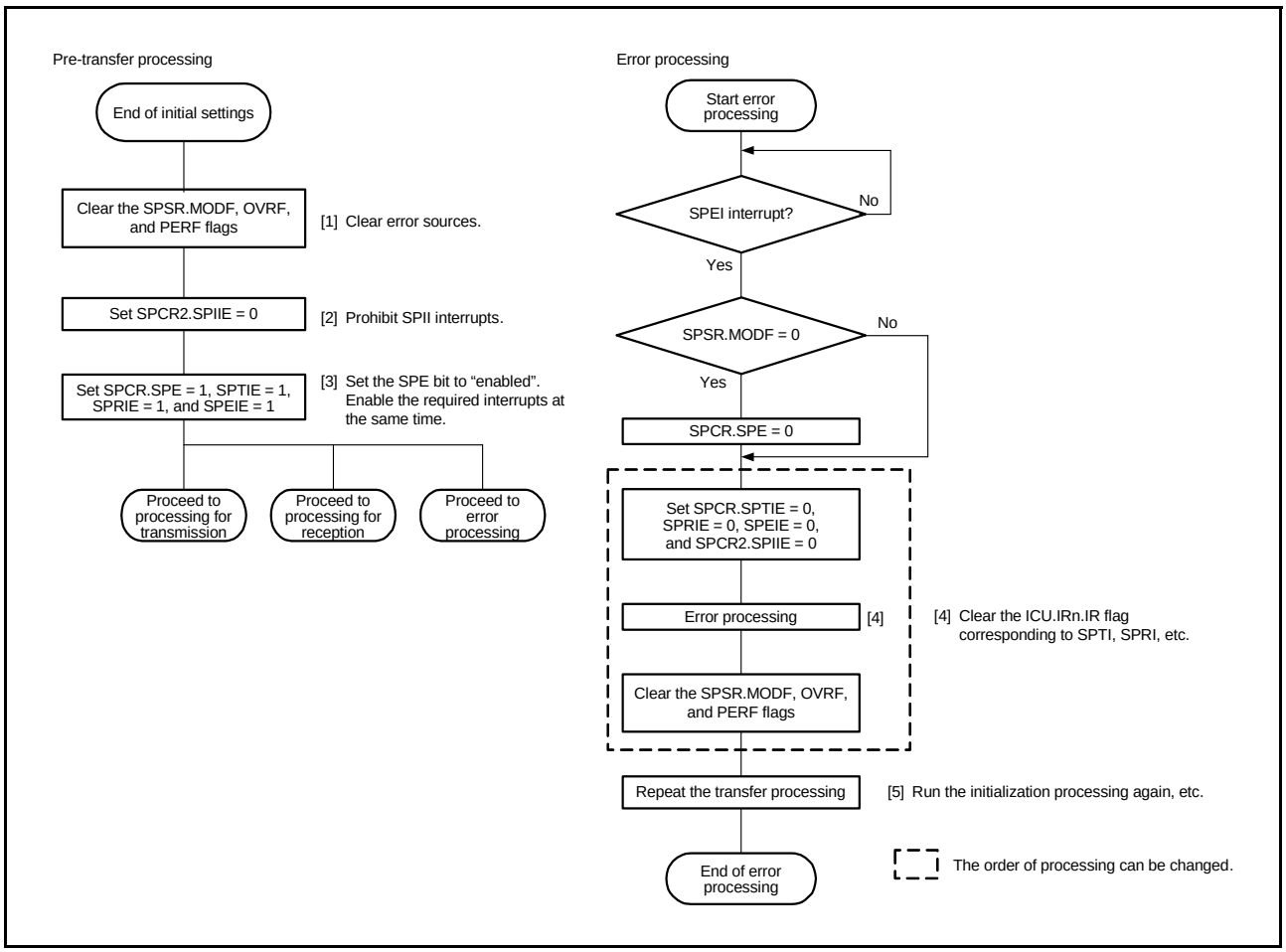



**Figure 28.40 Flowchart for Slave Mode (Error Processing)**

### 28.3.11 Clock Synchronous Operation

Setting the SPMS bit in the RSPI control register (SPCR) to 1 selects clock synchronous operation of the RSPI. In clock synchronous operation, the SSLAi pin is not used, and the three pins of RSPCKA, MOSIA, and MISOA handle communications. The SSLAi pin is available as I/O port pins.
Although clock synchronous operation does not require use of the SSLAi pin, operation of the module is the same as in SPI operation. That is, in both master and slave operations, communications can be performed with the same flow as in SPI operation. However, mode fault errors are not detected because the SSLAi pin is not used.
Furthermore, operation should not be performed if clock synchronous operation proceeds when the SPCMDm.CPHA bit is set to 0 in slave mode (SPCR.MSTR = 0).

#### 28.3.11.1 Master Mode Operation

##### (1) Starting a Serial Transfer

The RSPI updates the data in the transmit buffer (SPTX) of SPDR when data is written to the RSPI data register (SPDR) with the transmit buffer being empty (data for the next transfer is not set). When the shift register is empty after the number of frames set in the SPDCR.SPFC[1:0] bits are written to the SPDR, the RSPI copies data from the transmission buffer to the shift register and starts serial transmission. Upon copying transmit data to the shift register, the RSPI changes the status of the shift register to “full”, and upon termination of serial transfer, it changes the status of the shift register to “empty”. The status of the shift register cannot be referenced.
For details on the RSPI transfer format, refer to section 28.3.5, Transfer Format.
However, transfer in clock synchronous operation is conducted without the SSLA0 output signal.

##### (2) Terminating a Serial Transfer

The RSPI terminates the serial transfer after transmitting an RSPCKA edge corresponding to the sampling timing. If free space is available in the receive buffer, upon termination of serial transfer, the RSPI copies data from the shift register to the receive buffer of the RSPI data register (SPDR).
It should be noted that the final sampling timing varies depending on the bit length of transfer data. In master mode, the RSPI data length depends on the SPCMDm.SPB[3:0] bit setting.
For details on the RSPI transfer format, refer to section 28.3.5, Transfer Format.
However, transfer in clock synchronous operation is conducted without the SSLA0 output signal.

##### (3) Sequence Control

The transfer format employed in master mode is determined by SPSCR, SPCMDm, SPBR, SPCKD, SSLND, and SPND registers. Although the SSLAi signals are not output in clock synchronous operation, these settings are valid.
SPSCR is a register used to determine the sequence configuration for serial transfers that are executed by the RSPI in master mode. The following items are set in SPCMDm register: SSLAi output signal value, MSB/LSB first, data length, some of the bit rate settings, RSPCKA polarity/phase, whether SPCKD is to be referenced, whether SSLND is to be referenced, and whether SPND is to be referenced. SPBR holds some of the bit rate settings; SPCKD, an RSPI clock delay value; SSLND, an SSL negation delay; and SPND, a next-access delay value.
According to the sequence length that is assigned to SPSCR, the RSPI makes up a sequence comprised of a part or all of SPCMDm register. The RSPI contains a pointer to the SPCMDm register that makes up the sequence. The value of this pointer can be checked by reading the SPSSR.SPCP[2:0] bits. When the SPCR.SPE bit is set to 1 and the RSPI function is enabled, the RSPI loads the pointer to the commands in SPCMD0 register, and incorporates the SPCMD0 register setting into the transfer format at the beginning of serial transfer. The RSPI increments the pointer each time the next-access delay period for a data transfer ends. Upon completion of the serial transfer that corresponds to the final command comprising the sequence, the RSPI sets the pointer in SPCMD0 register, and in this manner the sequence is executed repeatedly.

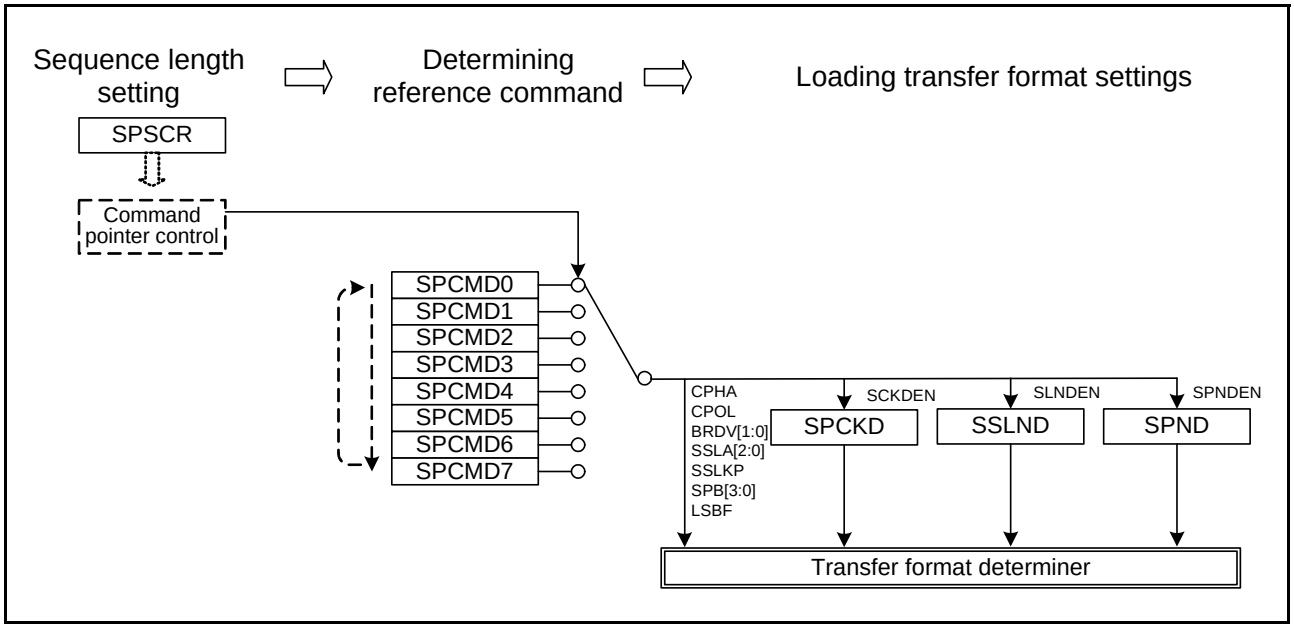



**Figure 28.41 Procedure for Determining the Form of Serial Transmission in Master Mode**

In this section, a frame is the combination of the data (SPDR) and the settings (SPCMDm).

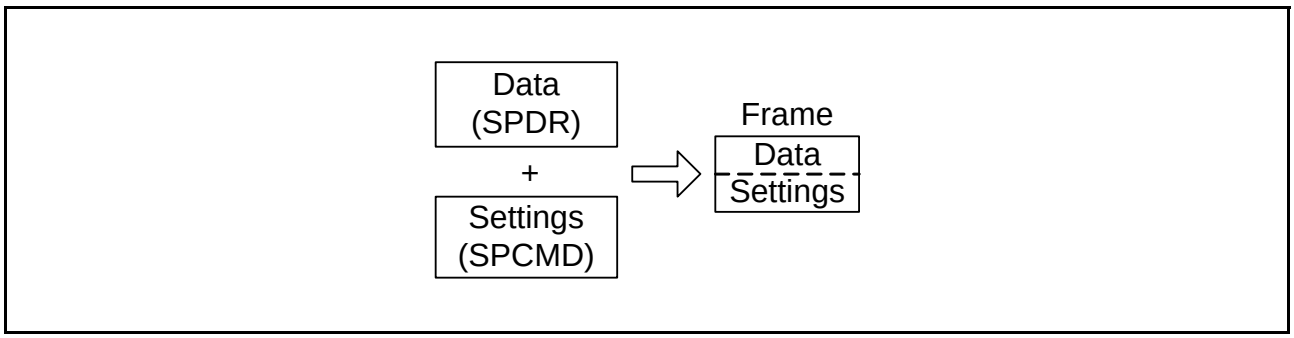



**Figure 28.42 Concept of a Frame**

Figure 28.43 shows the relationship between the command and the transmit and receive buffers in the sequence of operations specified by the settings in Table 28.4.

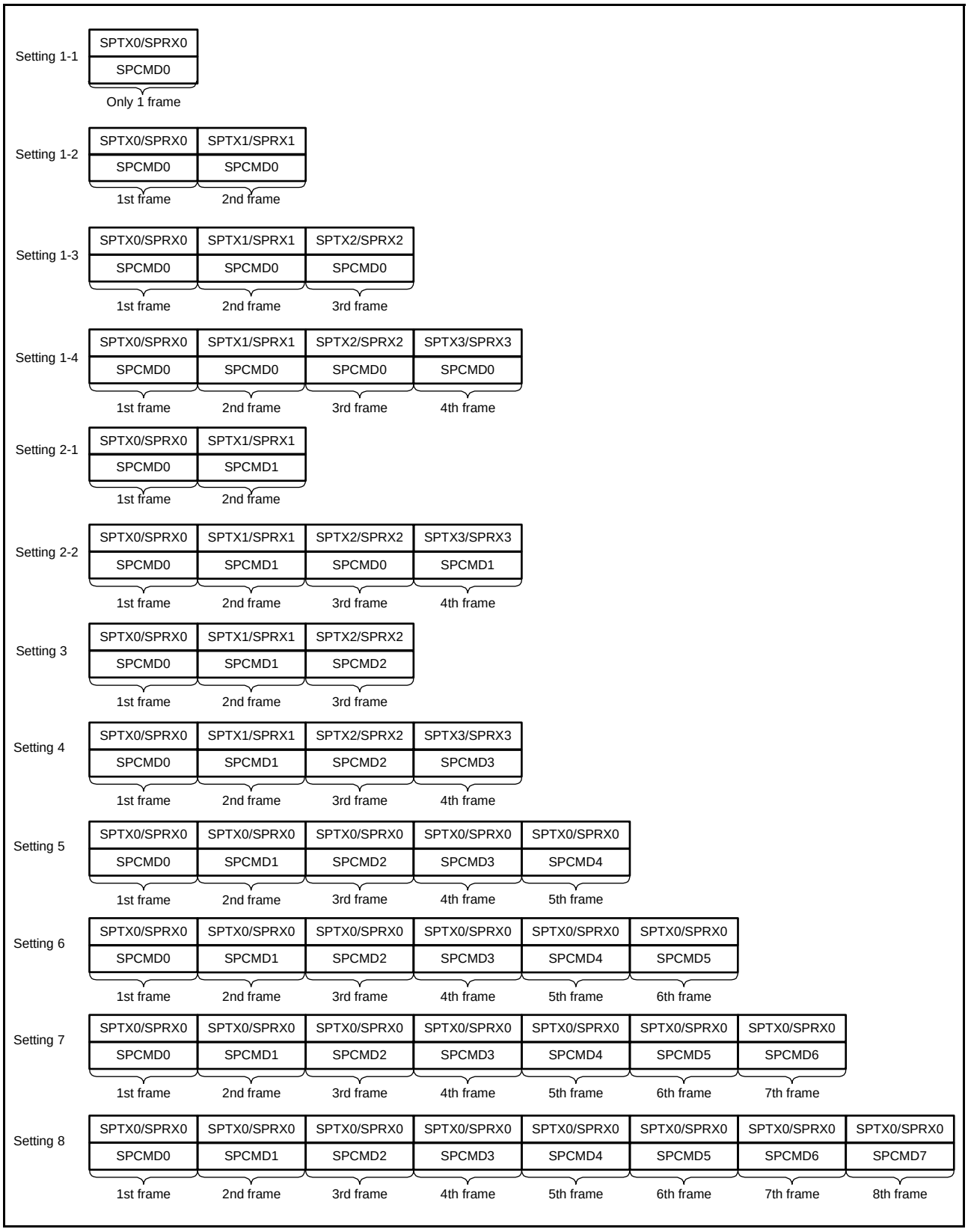



**Figure 28.43 Correspondence between the RSPI Command Register and Transmit/Receive Buffers in Sequence Operations**

### (4) Initialization Flowchart

Figure 28.44 is a flowchart illustrating an example of initialization in clock synchronous operation when the RSPI is used in master mode. For a description of how to set up the interrupt controller and I/O ports, refer to the descriptions given in the individual blocks.

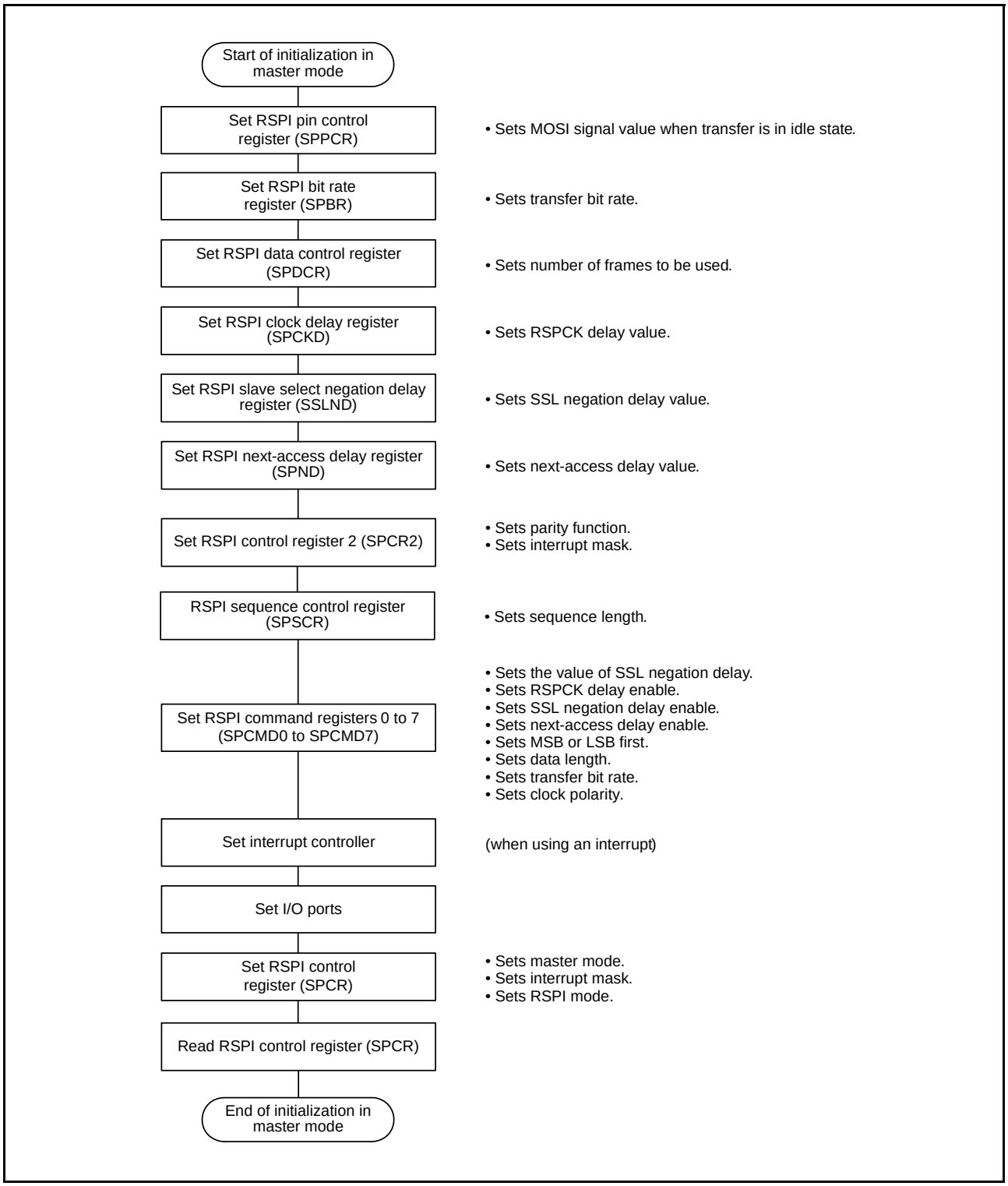



**Figure 28.44 Example of Initialization Flowchart in Master Mode (Clock Synchronous Operation)**

#### (5) Flow of Software Processing

Software processing during clock-synchronous master operation is the same as that for SPI master operation. For details, refer to section 28.3.10.1, (9) Software Processing Flow. Note that mode-fault errors will not occur.

### 28.3.11.2 Slave Mode Operation

#### (1) Starting a Serial Transfer

When the SPCR.SPMS bit is 1, the first RSPCKA edge triggers the start of a serial transfer in the RSPI.
When detecting the start of a serial transfer in a condition in which the shift register is empty, the RSPI changes the status of the shift register to “full”, so that data cannot be copied from the transmit buffer to the shift register when serial transfer is in progress. If the shift register was full before the serial transfer started, the RSPI keeps the status of the shift register unchanged, in the full state.
When the SPMS bit is 1, the RSPI always drives the MISOA output signal.
For details on the RSPI transfer format, refer to section 28.3.5, Transfer Format.
It should be noted that the SSL0 input signal is not used in clock synchronous operation.

#### (2) Terminating a Serial Transfer

The RSPI terminates the serial transfer after detecting an RSPCKA edge corresponding to the final sampling timing.
When free space is available in the receive buffer, upon termination of serial transfer the RSPI copies received data from the shift register to the receive buffer of the RSPI data register (SPDR). Upon termination of a serial transfer the RSPI changes the status of the shift register to “empty”. The final sampling timing changes depending on the bit length of transfer data. In slave mode, the RSPI data length depends on the SPCMD0.SPB[3:0] bit setting.
For details on the RSPI transfer format, refer to section 28.3.5, Transfer Format.

### (3) Initialization Flowchart

Figure 28.45 is a flowchart illustrating an example of initialization in clock synchronous operation when the RSPI is used in slave mode. For a description of how to set up the interrupt controller and I/O ports, refer to the descriptions given in the individual blocks.

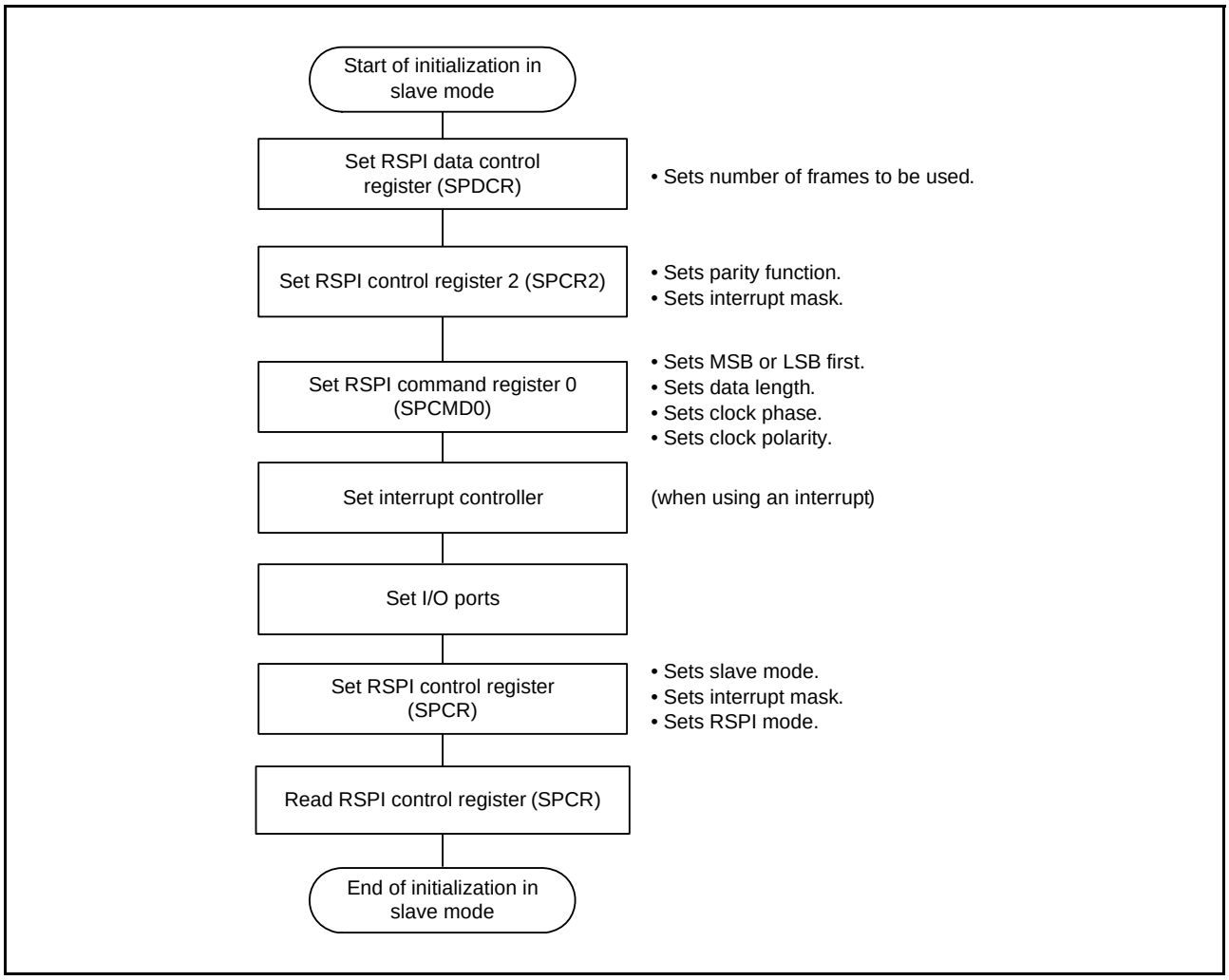



**Figure 28.45 Example of Initialization Flowchart in Slave Mode (Clock Synchronous Operation)**

### (4) Flow of Software Processing

Software processing during clock-synchronous slave operation is the same as that for SPI slave operation. For details, refer to section 28.3.10.2, (6) Software Processing Flow. Note that mode-fault errors will not occur.

### 28.3.12 Loopback Mode

When 1 is written to the SPPCR.SPLP2 bit or SPPCR.SPLP bit, the RSPI shuts off the path between the MISOA pin and the shift register if the SPCR.MSTR bit is 1, and between the MOSIA pin and the shift register if the SPCR.MSTR bit is 0, and connects the input path and output path of the shift register. The RSPI does not shut off the path between the MOSIA pin and the shift register if the SPCR.MSTR bit is 1, and between the MISOA pin and the shift register if the SPCR.MSTR bit is 0. This is called loopback mode. When a serial transfer is executed in loopback mode, the transmit data for the RSPI or the reversed transmit data becomes the received data for the RSPI.
Table 28.12 lists the relationship among the SPLP2 and SPLP bits and the received data. Figure 28.46 shows the configuration of the shift register I/O paths for the case where the RSPI in master mode is set in loopback mode (SPPCR.SPLP2 = 0, SPPCR.SPLP = 1).

**Table 28.12 SPLP2 and SPLP Bit Settings and Received Data**

| SPPCR.SPLP2 Bit | SPPCR.SPLP Bit | Received Data |
|---|---|---|
| 0 | 0 | Input data from the MOSIA pin or MISOA pin |
| 0 | 1 | Reversed transmit data |
| 1 | 0 | Transmit data |
| 1 | 1 | Transmit data |

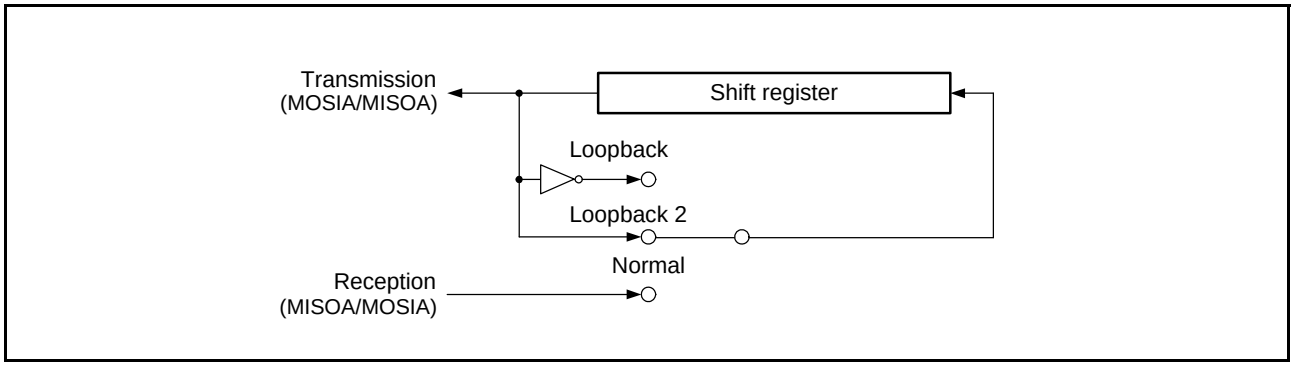



**Figure 28.46 Configuration of Shift Register I/O Paths in Loopback Mode (Master Mode)**

### 28.3.13 Self-Diagnosis of Parity Bit Function

The parity circuit consists of a parity bit adding unit used for transmit data and an error detecting unit used for received data. In order to detect defects in the parity bit adding unit and error detecting unit of the parity circuit, self-diagnosis is executed for the parity circuit following the flowchart shown in Figure 28.47.

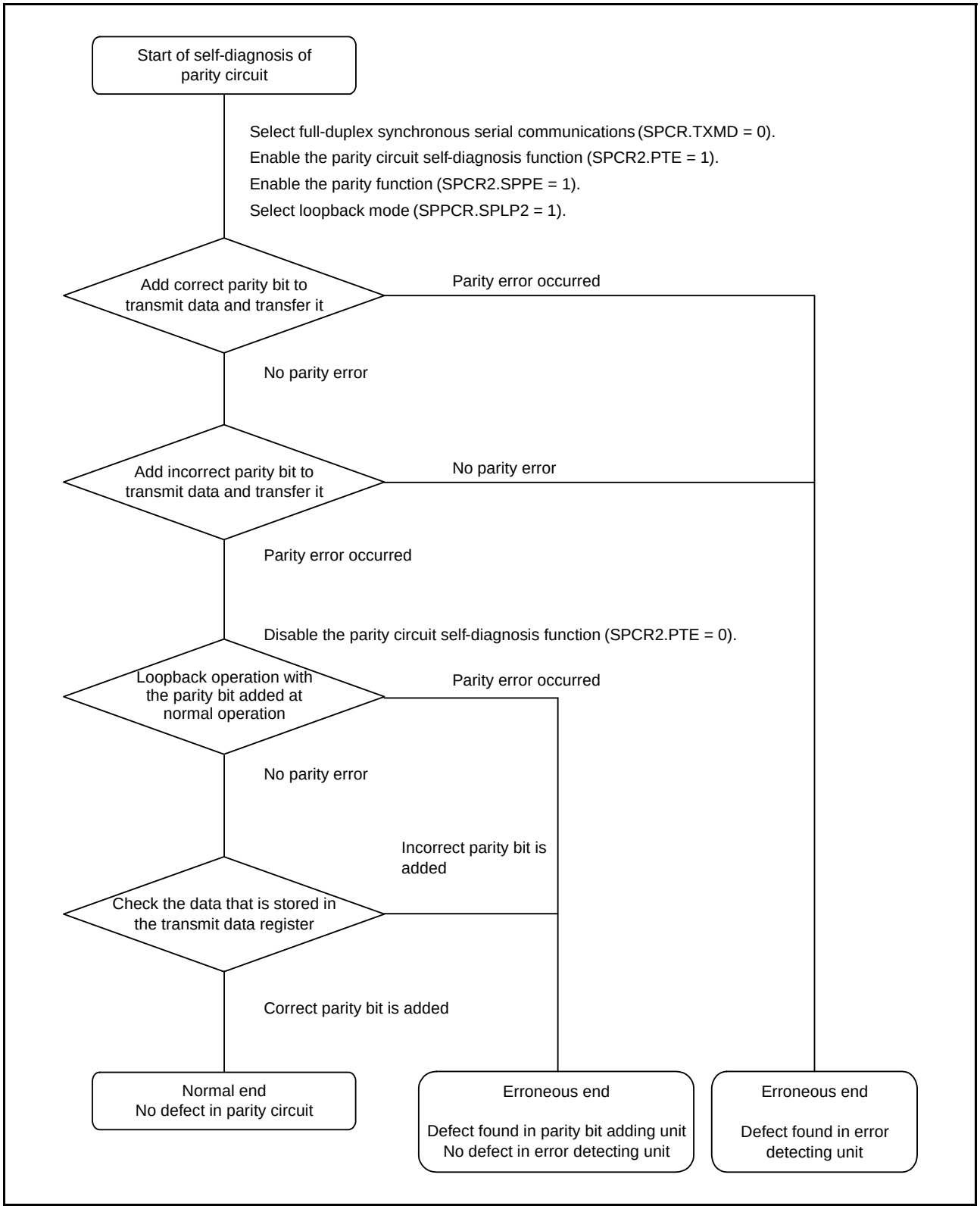



**Figure 28.47 Flowchart for Self-Diagnosis of Parity Circuit**

### 28.3.14 Interrupt Sources

The RSPI has interrupt sources of receive buffer full, transmit buffer empty, mode fault, overrun, parity error, and RSPI idle. In addition, the DTC can be activated by the receive buffer full or transmit buffer empty interrupt to perform data transfer.

Since the vector address for SPEI is allocated to interrupt requests due to mode-fault, overrun, and parity errors, the actual interrupt source must be determined from the flags. Interrupt sources for the RSPI are listed in Table 28.13. An interrupt is generated on satisfaction of an interrupt condition in Table 28.13. Clear the receive buffer full and transmit buffer empty sources through data transfer.

When using the DTC to perform data transmission/reception, the DTC must be set up first to be in a status in which transfer is enabled before making the RSPI settings. For the method for setting the DTC, refer to section 16, Data Transfer Controller (DTCa).

If the conditions for generating a transmit buffer empty or receive buffer full interrupt are generated while the ICU.IRn.IR flag is 1, the interrupt is not output as a request for ICU but is retained internally (the capacity for retention is one request per source). A retained interrupt request is output when the ICU.IRn.IR flag becomes 0. A retained interrupt request is automatically discarded once it is output as an actual interrupt request. The interrupt enable bit (the SPCR.SPTIE or SPCR.SPRIE bit) for an internally retained interrupt request can also be cleared to 0.

**Table 28.13 Interrupt Sources of RSPI**

| Interrupt Source | Symbol | Interrupt Condition | DTC Activation |
|---|---|---|---|
| Receive buffer full | SPRI | The receive buffer becomes full while the SPCR.SPRIE bit is 1. | Possible |
| Transmit buffer empty | SPTI | The transmit buffer becomes empty while the SPCR.SPTIE bit is 1. | Possible |
| RSPI errors (mode fault, overrun and parity error) | SPEI | The SPSR.MODF, OVRF, or PERF flag is set to 1 while the SPCR.SPEIE bit is 1. | Impossible |
| RSPI idle | SPII | The SPSR.IDLNF flag is set to 0 while the SPCR2.SPIIE bit is 1. | Impossible |

## 28.4 Usage Note

### 28.4.1 Setting Module Stop Function

Module stop control register B (MSTPCRB) can used to enable or disable operation of the RSPI. The RSPI is stopped after a reset. The registers become accessible on release from the module-stop state. For details, refer to section 11, Low Power Consumption.

### 28.4.2 Cautionary Note on the Low Power Consumption Functions

When a low power consumption function is to be used to lower power consumption by the RSPI, use the low power consumption function after the SPCR.SPE bit is set to 0 and transfer ends.

### 28.4.3 Points to Note on Starting Transfer

If the ICU.IRn.IR flag is 1 at the time transfer is to be started, an interrupt request is internally retained after transfer starts, and this can lead to unanticipated behavior of the ICU.IRn.IR flag.
When the ICU.IRn.IR flag is 1 at the time transfer is to start, follow the procedure below to clear interrupt requests before enabling operations (by setting the SPCR.SPE bit to 1).

1. Confirm that transfer has stopped (i.e. that the SPCR.SPE bit is 0).
2. Clear the relevant interrupt enable bit (the SPCR.SPTIE or SPCR.SPRIE bit) to 0.
3. Read the relevant interrupt enable bit (the SPCR.SPTIE or SPCR.SPRIE bit) and confirm that its value is 0.
4. Clear the ICU.IRn.IR flag to 0.

# 29. CRC Calculator (CRC)

The CRC (Cyclic Redundancy Check) calculator generates CRC codes.

## 29.1 Overview

Table 29.1 lists the specifications of the CRC calculator, and Figure 29.1 shows a block diagram of the CRC calculator.

**Table 29.1 Specifications of the CRC Calculator**

| Item | Description |
|---|---|
| Data for CRC calculation*1 | CRC code generated for any desired data in 8n-bit units (where n is a whole number) |
| CRC processor unit | Operation executed on 8 bits in parallel |
| CRC generating polynomial | One of three generating polynomials selectable<br>• 8-bit CRC<br>$X^8 + X^2 + X + 1$<br>• 16-bit CRC<br>$X^{16} + X^{15} + X^2 + 1$<br>$X^{16} + X^{12} + X^5 + 1$ |
| CRC calculation switching | The bit order of CRC calculation results can be switched for LSB first or MSB first communication |
| Low-power consumption function | Module stop state can be set |

Note 1. The circuit does not have functionality to divide data for calculation into CRC calculation units. Write data in 8-bit units.

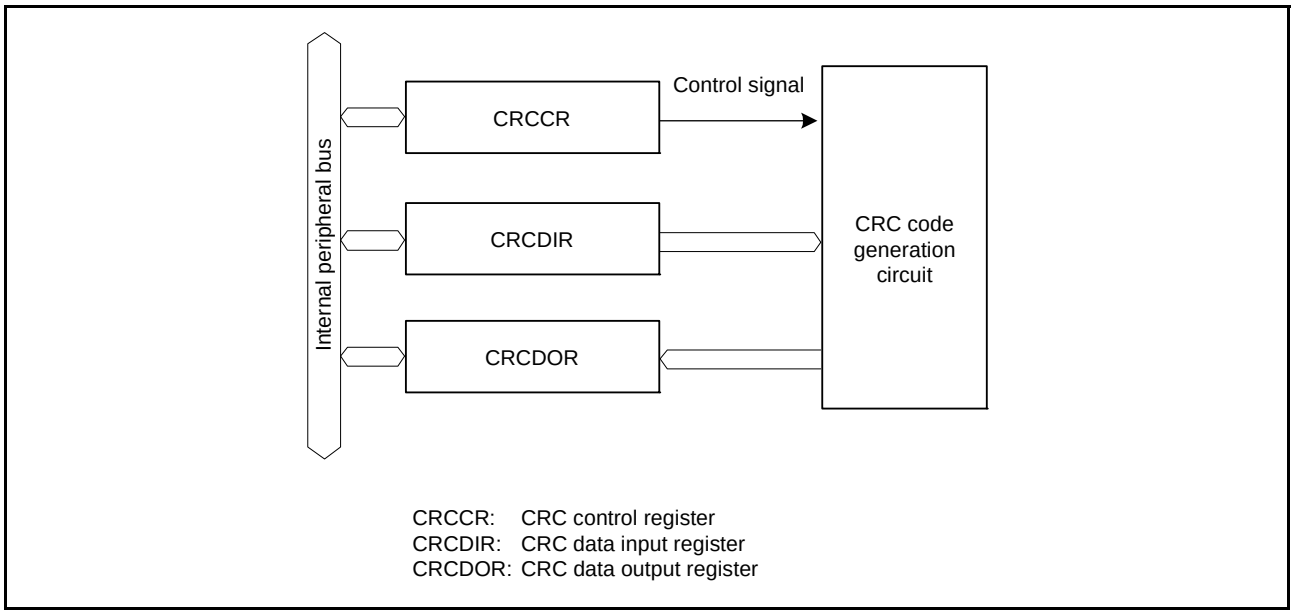



**Figure 29.1 Block Diagram of CRC Calculator**

## 29.2 Register Descriptions

### 29.2.1 CRC Control Register (CRCCR)

Address(es): 0008 8280h

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | DORCLR | — | — | — | — | LMS | GPS[1:0] | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b1, b0 | GPS[1:0] | CRC Generating Polynomial Switching | b1 b0<br>0 0: No calculation is executed.<br>0 1: 8-bit CRC ($X^8 + X^2 + X + 1$)<br>1 0: 16-bit CRC ($X^{16} + X^{15} + X^2 + 1$)<br>1 1: 16-bit CRC ($X^{16} + X^{12} + X^5 + 1$) | R/W |
| b2 | LMS | CRC Calculation Switching | 0: Generates CRC for LSB first communication<br>1: Generates CRC for MSB first communication | R/W |
| b6 to b3 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |
| b7 | DORCLR | CRCDOR Register Clear | 1: Clear the CRCDOR register<br>This bit is read as 0. | W*1 |

Note 1. Only 1 can be written.

#### DORCLR Bit (CRCDOR Register Clear)

Write 1 to this bit so that the CRCDOR register is cleared to 0000h.
This bit is read as 0. Only 1 can be written.

#### LMS Bit (CRC Calculation Switching)

Set this bit to select the bit order of generated 16-bit CRC code. Transmit the lower-order byte (bits 7 to 0) of the CRC code first for LSB first communication and the higher-order byte (bits 15 to 8) first for MSB first communication. For details on transmitting and receiving CRC code, refer to section 29.3, Operation.

### 29.2.2 CRC Data Input Register (CRCDIR)

Address(es): 0008 8281h

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

CRCDIR is a readable/writable register. Write data for CRC calculation to this register.

### 29.2.3 CRC Data Output Register (CRCDOR)

Address(es): 0008 8282h

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

CRCDOR is a readable/writable register.
Since its initial value is 0000h, rewrite the CRCDOR register to perform calculation using a value other than the initial value.
Data written to the CRCDIR register is CRC calculated and the result is stored in the CRCDOR register. If the CRC code is calculated following the transferred data and the result is 0000h, there is no CRC error.
When an 8-bit CRC ($X^8 + X^2 + X + 1$ polynomial) is in use, the valid CRC code is obtained in the low-order byte (b7 to b0). The high-order byte (b15 to b8) is not updated.

## 29.3 Operation

The CRC calculator generates CRC codes for use in LSB first or MSB first transfer.
The following shows examples of generating the CRC code for input data (F0h) using the 16-bit CRC generator polynomial ($X^{16} + X^{12} + X^5 + 1$). In these examples, the value of the CRC data output register (CRCDOR) is cleared before CRC calculation.
When an 8-bit CRC (with the polynomial $X^8 + X^2 + X + 1$) is in use, the valid bits of the CRC code are obtained in the lower-order byte of CRCDOR.

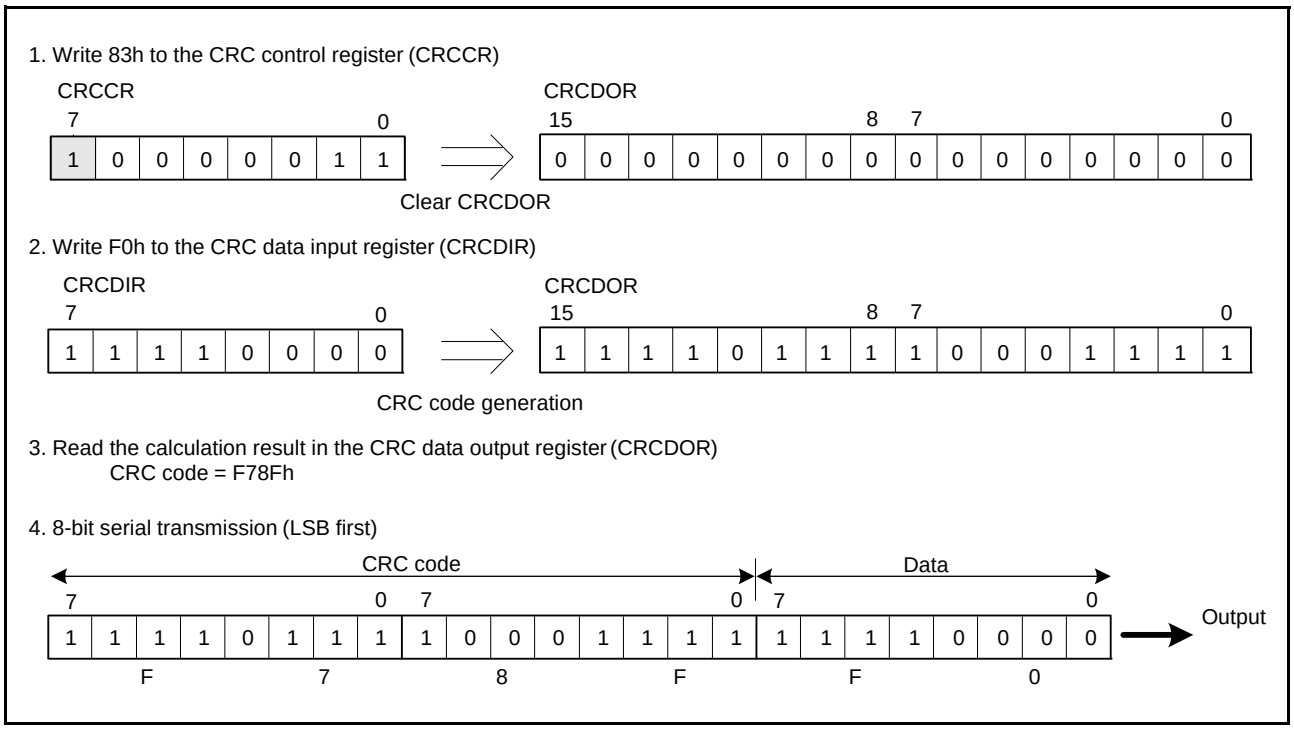



**Figure 29.2 LSB First Data Transmission**

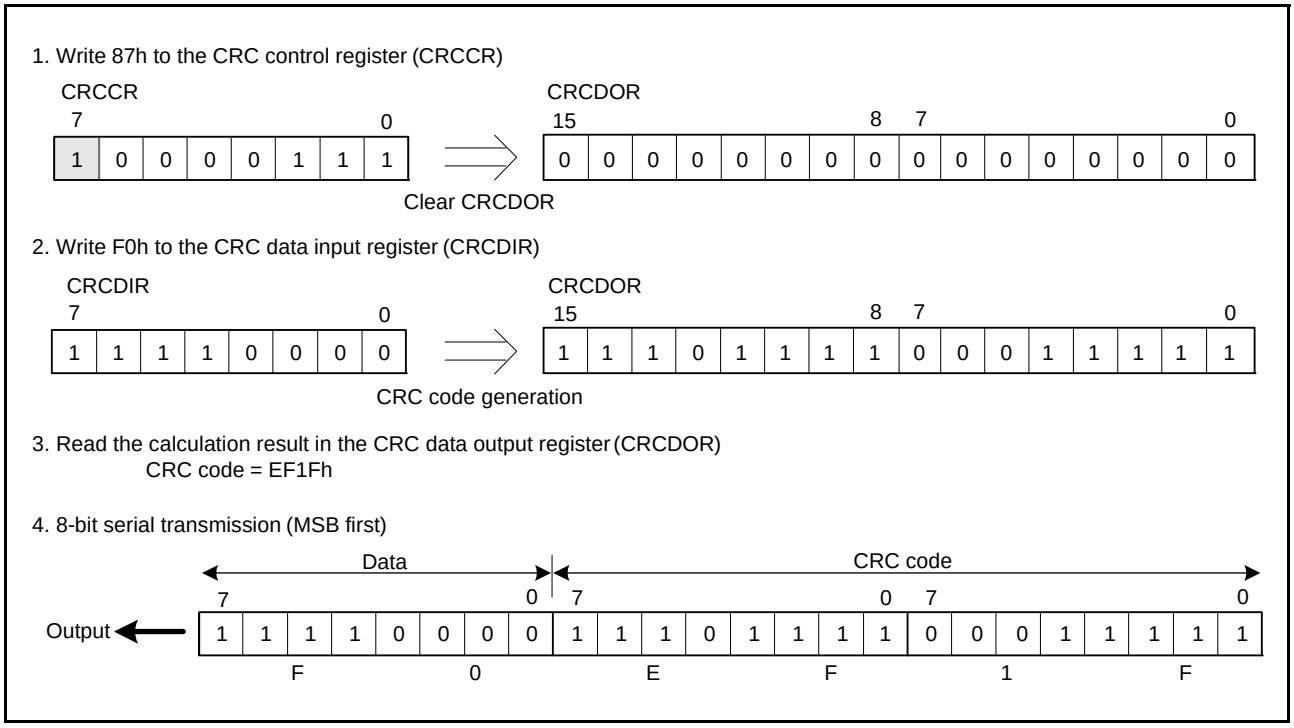



**Figure 29.3 MSB-First Data Transmission**

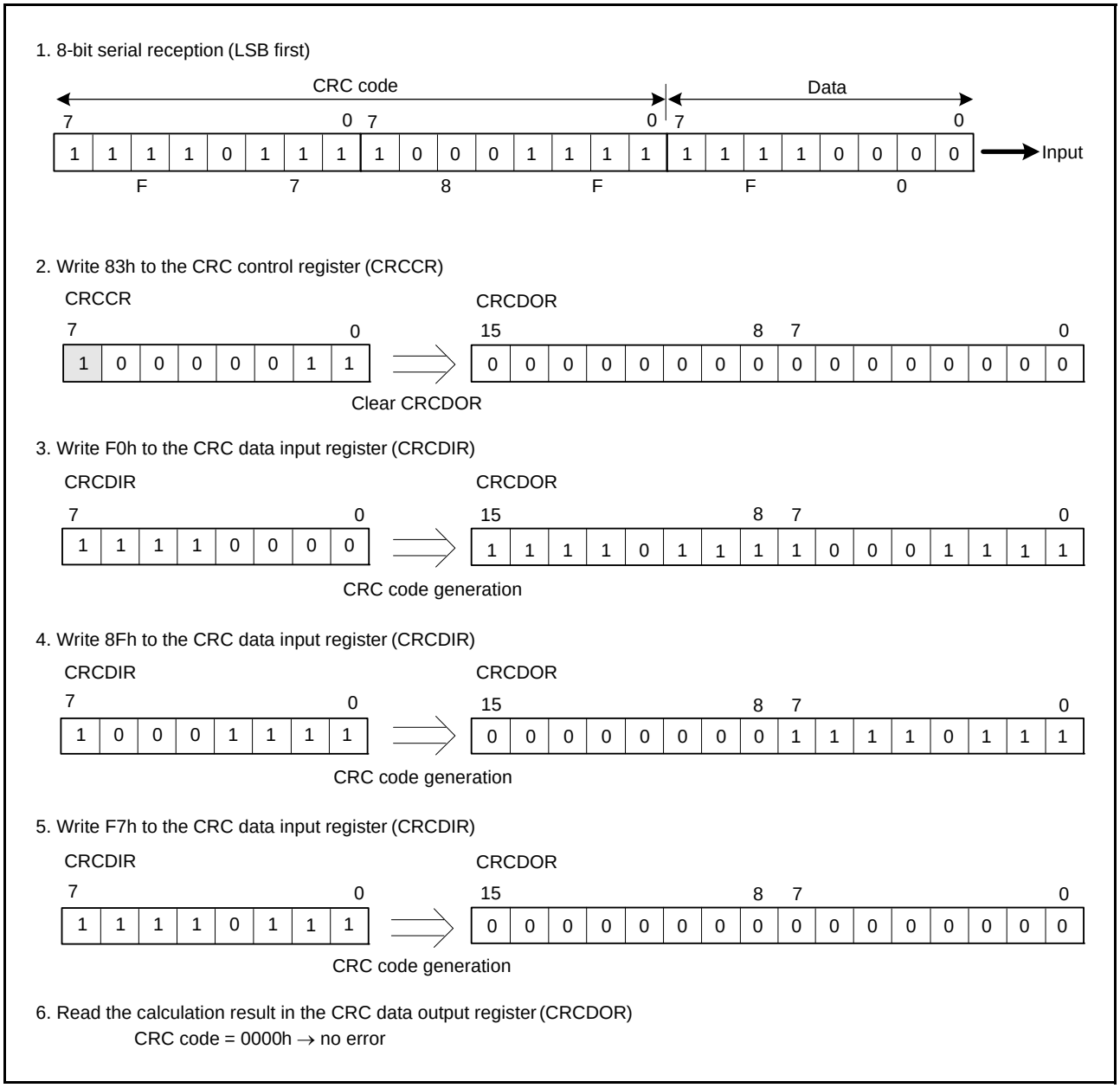



**Figure 29.4 LSB First Data Reception**

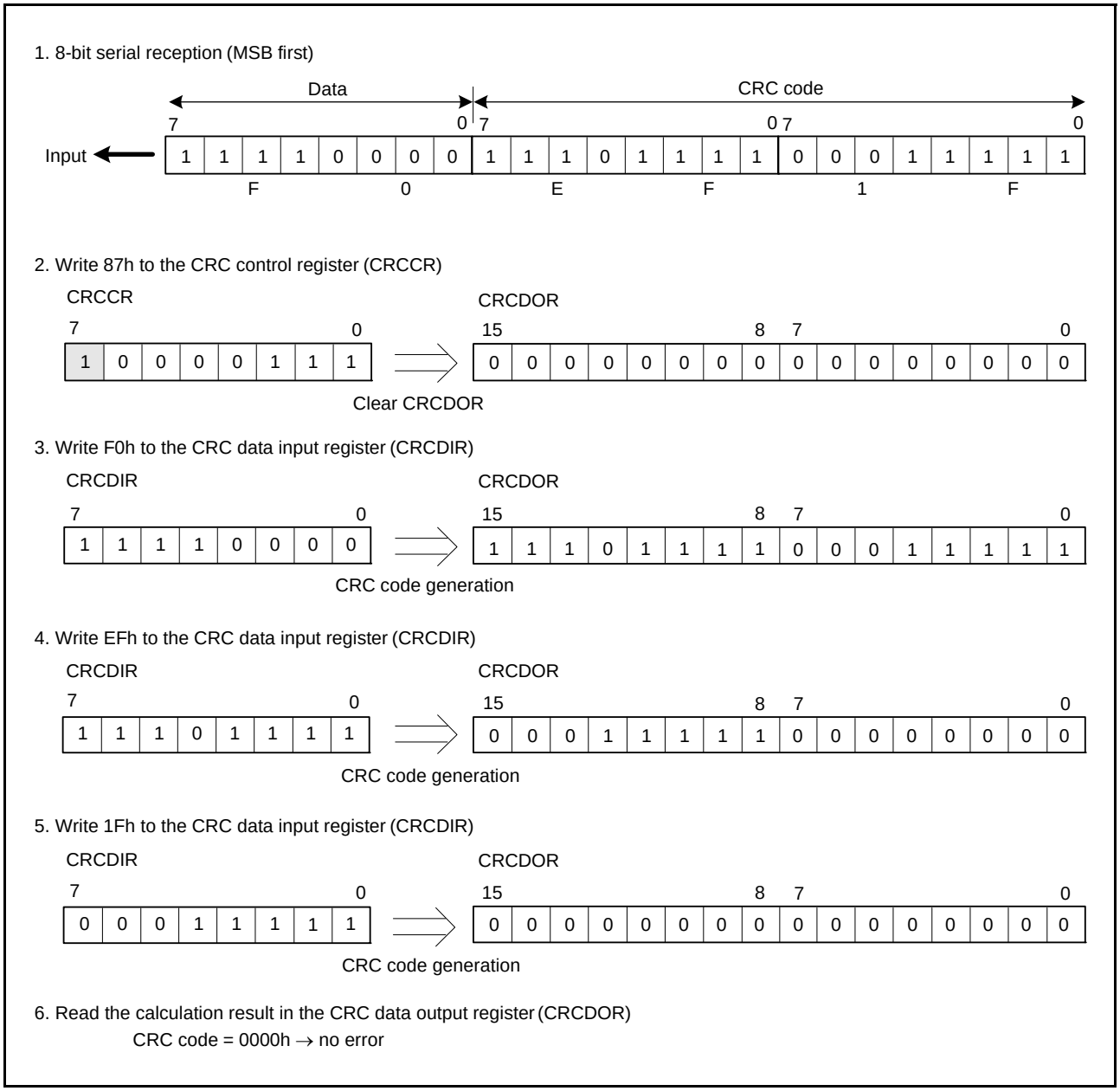



**Figure 29.5 MSB First Data Reception**

## 29.4 Usage Notes

### 29.4.1 Module Stop Function Setting

Operation of the CRC calculator can be disabled or enabled using module stop control register B (MSTPCRB). After a reset, the CRC is in the module stop state. Register access is enabled by clearing the module stop state. For details, see section 11, Low Power Consumption.

### 29.4.2 Note on Transmission

Note that the sequence of transmission for the CRC code differs according to whether transmission is LSB first or MSB first.

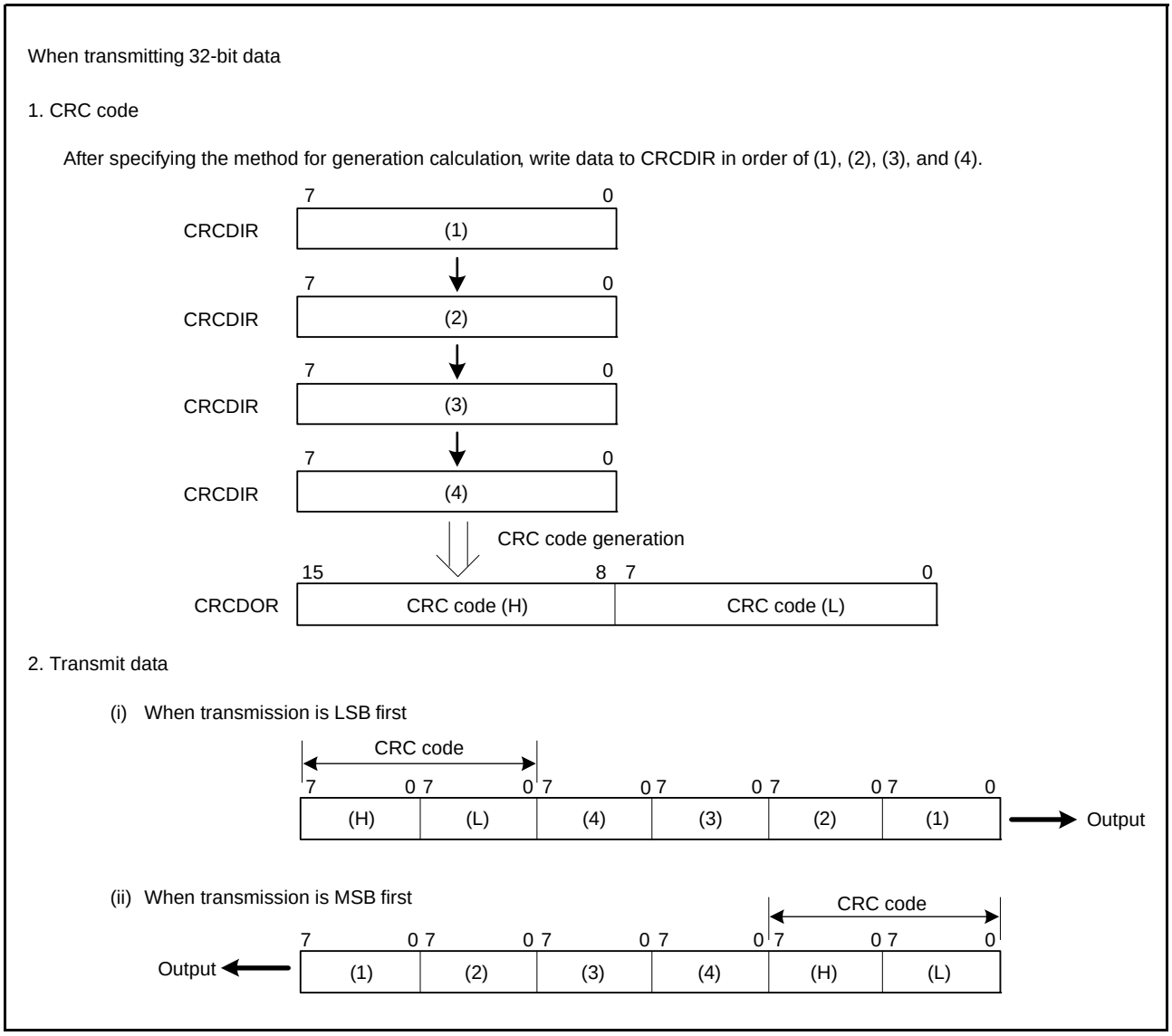



**Figure 29.6 LSB First and MSB First Data Transmission**

# 30. 12-Bit A/D Converter (S12ADb)

## 30.1 Overview

This MCU includes a 12-bit successive approximation A/D converter. Up to 14 channel analog inputs, temperature sensor outputs, or internal reference voltages can be selected.
The 12-bit A/D converter converts a maximum of 14 selected channels of analog inputs, temperature sensor outputs, or internal reference voltages into a 12-bit digital value through successive approximation.
The A/D converter has three operating modes: single scan mode in which the analog inputs of up to 14 arbitrarily selected channels are converted for only once in ascending channel order; and continuous scan mode in which the analog inputs of up to 14 arbitrarily selected channels are continuously converted in ascending channel order; and group scan mode in which up to 14 channels of the analog inputs are arbitrarily divided into two groups (group A and group B) and converted in ascending channel order in each group.
In group scan mode, the scan start conditions of group A and group B can be independently selected, thus allowing A/D conversion of group A and group B to be started independently.
In double trigger mode, one arbitrarily selected analog input channel is converted in single scan mode or group scan mode (group A), and the resulting data of A/D conversion started by the first and second triggers are stored into separate registers (duplication of A/D conversion data).
A/D conversion of the temperature sensor output or the internal reference voltage is accomplished independently.
The VREFH0 or AVCC0 pin can be selected as the reference power supply pin for high-electric potential. The VREFL0 or AVSS0 pin can be selected as the reference power supply ground pin for low-electric potential.
Table 30.1 lists the specifications of the 12-bit A/D converter and Table 30.2 indicates the functions of the 12-bit A/D converter. Figure 30.1 shows a block diagram of the 12-bit A/D converter.

**Table 30.1 Specifications of 12-Bit A/D Converter (1/2)**

| Item | Specifications |
|---|---|
| Number of units | One unit |
| Input channels | Up to 14 channels |
| Extended analog inputs | Temperature sensor output, internal reference voltage |
| A/D conversion method | Successive approximation method |
| Resolution | 12 bits |
| Conversion time | 1.0 µs per channel (when A/D conversion clock ADCLK = 32 MHz) |
| A/D conversion clock | Peripheral module clock PCLK*1 and A/D conversion clock ADCLK*1 can be set so that the frequency division ratio should be one of the following.<br>PCLK to ADCLK frequency division ratio = 1:1, 1:2, 1:4, 1:8, 2:1, 4:1<br>ADCLK is set using the clock generation circuits. |
| Data registers | For analog input: 14 data registers<br>For duplication of A/D conversion data in double trigger mode: One data register<br>For temperature sensor: One data register<br>For internal reference voltage: One data register<br>The A/D conversion result is stored in 12-bit A/D data registers.<br>In addition mode, A/D conversion results are added and stored in A/D data registers as 14-bit data.<br>Duplication of A/D conversion data<br>• A/D conversion data of one selected analog input channel is stored into A/D data register y when conversion is started by the first trigger and into the duplication register when started by the second trigger.<br>• Duplication is available only in double trigger mode in single scan mode or group scan mode. |

**Table 30.1 Specifications of 12-Bit A/D Converter (2/2)**

| Item | Specifications |
|---|---|
| Operating modes | • Single scan mode:<br>A/D conversion is performed for only once on the analog inputs of up to 14 arbitrarily selected channels.<br>A/D conversion is performed only once on the temperature sensor output.<br>A/D conversion is performed only once on the internal reference voltage.<br>• Continuous scan mode: A/D conversion is performed repeatedly on the analog inputs of up to 14 arbitrarily selected channels.*2<br>• Group scan mode:<br>Up to 14 channels of analog inputs are divided into group A and group B and A/D conversion is performed only once on all the selected channels on a group basis. The scan start conditions of group A and group B can be independently selected, thus allowing A/D conversion of group A and group B to be started independently. |
| Conditions of A/D conversion start | • Software trigger<br>• Synchronous trigger<br>Trigger by MTU or ELC<br>• Asynchronous trigger<br>A/D conversion can be triggered from the ADTRG0# pin. |
| Functions | • Variable sampling state count<br>• A/D-converted value addition mode<br>• Double trigger mode (duplication of A/D conversion data) |
| Interrupt sources | • In the modes except double trigger mode and group scan mode, A/D scan end interrupt (S12ADI0) request can be generated on completion of single scan.<br>• In double trigger mode, A/D scan end interrupt (S12ADI0) request can be generated on completion of double scan.<br>• In group scan mode, A/D scan end interrupt (S12ADI0) request can be generated on completion of group A scan, whereas A/D scan end interrupt specially for group B (GBADI) request can be generated on completion of group B scan.<br>• In group scan mode with double trigger mode, A/D scan end interrupt (S12ADI0) request can be generated on completion of double scan of group A, whereas A/D scan end interrupt specially for group B (GBADI) request can be generated on completion of group B scan.<br>• S12ADI0 or GBADI interrupt can activate data transfer controller (DTC). |
| Event linking | • An ELC event can be generated on completion of scans except for group B scan in group scan mode.<br>• A/D conversion can be started by the trigger from ELC. |
| Reference voltage | • Set the PJ6PFS.ASEL bit to select the VREFH0 or AVCC0 pin as the reference power supply pin for high-electric potential.<br>• Set the PJ7PFS.ASEL bit to select the VREFL0 or AVSS0 pin as the reference power supply ground pin for low-electric potential. |
| Low power consumption function | • Module stop state can be specified.*3 |

Note 1. Peripheral module clock PCLK is set according to the setting of SCKCR.PCKB[3:0] and A/D conversion clock ADCLK is set according to the setting of SCKCR.PCKD[3:0].
Note 2. Do not use continuous scan mode or group scan mode when temperature sensor output or internal reference voltage is selected.
Note 3. When the module stop state is canceled, A/D conversion can be started after 1 μs has elapsed.

**Table 30.2 Functions of 12-Bit A/D Converter**

| Item | | | Function |
|---|---|---|---|
| Analog input channels | | | AN000 to AN004, AN006, AN008 to AN015, temperature sensor output, internal reference voltage |
| A/D conversion start conditions | Software | Software trigger | Enabled |
| | Asynchronous trigger | ADTRG0# | Enabled |
| | Synchronous trigger | TRGA compare match/input capture from MTU0 | TRG0AN |
| | | TRGB compare match/input capture from MTU0 | TRG0BN |
| | | TRGA compare match/input capture or MTU4.TCNT underflow (trough) in complementary PWM mode from MTU0 to MTU4 | TRGAN |
| | | TRGE compare match from MTU0 | TRG0EN |
| | | TRGF compare match from MTU0 | TRG0FN |
| | | MTU4.TADCORA and MTU4.TCNT compare match (interrupt skipping function 1) | TRG4AN |
| | | MTU4.TADCORB and MTU4.TCNT compare match (interrupt skipping function 1) | TRG4BN |
| | | MTU4.TADCORA and MTU4.TCNT compare match and MTU4.TADCORB and MTU4.TCNT compare match (interrupt skipping function 1) | TRG4ABN |
| | | Trigger from ELC | Enabled |
| Interrupt | | | S12ADI0 interrupt, GBADI interrupt |
| Module stop function setting*1 | | | MSTPCRA.MSTPA17 bit |

Note 1. For details, refer to section 11, Low Power Consumption.

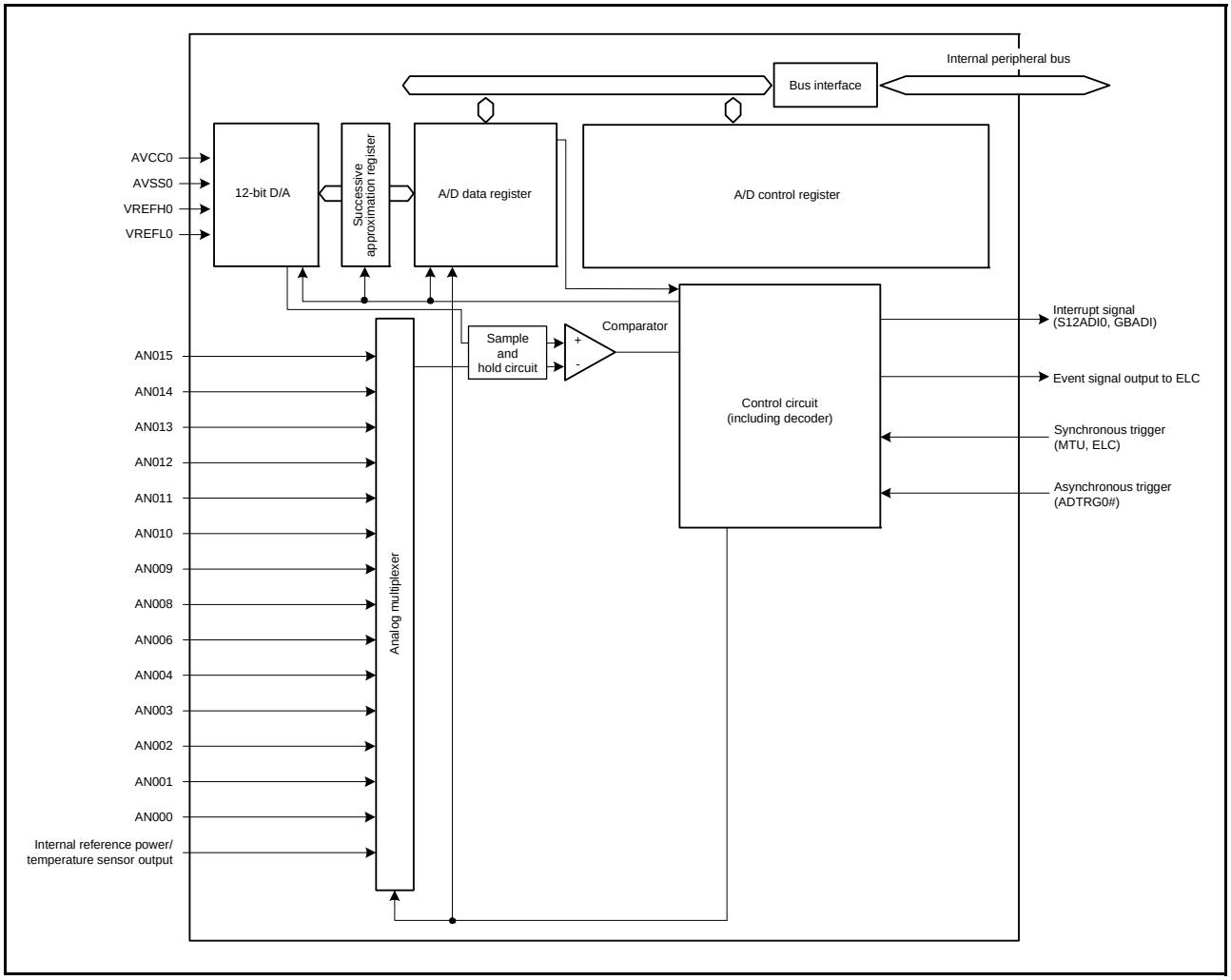



**Figure 30.1 Block Diagram of 12-Bit A/D Converter**

Table 30.3 lists the input pins of the 12-bit A/D converter.

**Table 30.3 Input Pins of 12-Bit A/D Converter**

| Pin Name | Input | Function |
|---|---|---|
| AVCC0 | Input | Analog block power supply pin |
| AVSS0 | Input | Analog block ground pin |
| VREFH0 | Input | Reference power supply pin |
| VREFL0 | Input | Reference ground pin |
| AN000 to AN004, AN006, AN008 to AN015 | Input | Analog input pins |
| ADTRG0# | Input | External trigger input pin for starting A/D conversion |

## 30.2 Register Descriptions

### 30.2.1 A/D Data Registers y (ADDRy) (y = 0 to 4, 6, 8 to 15)

ADDRy are 16-bit read-only registers which store the A/D conversion results of channels AN000 to AN004, AN006, AN008 to AN015.

The A/D data registers use the following different formats depending on the setting of the A/D data register format select bit (ADRFMT) in ADCER or A/D-converted value addition mode.

- ADCER.ADRFMT = 0 (Setting for right-alignment)

Address(es): ADDR0: 0008 9020h to ADDR4: 0008 9028h, ADDR6: 0008 902Ch, ADDR8: 0008 9030h to ADDR15: 0008 903Eh

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | AD[11:0] | | | | | | | | | | | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b11 to b0 | AD[11:0] | — | 12-bit A/D-converted value | R |
| b15 to b12 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |

- ADCER.ADRFMT = 1 (Setting for left-alignment)

Address(es): ADDR0: 0008 9020h to ADDR4: 0008 9028h, ADDR6: 0008 902Ch, ADDR8: 0008 9030h to ADDR15: 0008 903Eh

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | AD[11:0] | | | | | | | | | | | | — | — | — | — |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b3 to b0 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |
| b15 to b4 | AD[11:0] | — | 12-bit A/D-converted value | R |

- When A/D-converted value addition mode is selected

Address(es): ADDR0: 0008 9020h to ADDR4: 0008 9028h, ADDR6: 0008 902Ch, ADDR8: 0008 9030h to ADDR15: 0008 903Eh

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | AD[13:0] | | | | | | | | | | | | | | — | — |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b1, b0 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |
| b15 to b2 | AD[13:0] | — | 14-bit A/D-converted value addition result | R |

When A/D-converted value addition mode is selected, the AD[13:0] bits in ADDRy show the value added by the A/D-converted value of the respective channels. In A/D-converted value addition mode, the setting of the ADRFMT bit in ADCER becomes invalid and the format of the register becomes left-aligned.

The following minimum and maximum values apply to channels on which A/D-converted value addition mode is selected.

First conversion: 0000h ≤ ADDRy (y = 0 to 4, 6, 8 to 15) ≤ 3FFCh
(ADDRy (y = 0 to 4, 6, 8 to 15): Bits 15 and 14 = 00b, bits 13 to 2 = AD11 to AD0, bits 1 and 0 = 00b)
Second conversion: 0000h ≤ ADDRy (y = 0 to 4, 6, 8 to 15) ≤ 7FF8h
(ADDRy (y = 0 to 4, 6, 8 to 15): Bit 15 = 0b, bits 14 to 2 = AD12 to AD0, bits 1 and 0 = 00b)
Third conversion: 0000h ≤ ADDRy (y = 0 to 4, 6, 8 to 15) ≤ BFF4h
(ADDRy (y = 0 to 4, 6, 8 to 15): Bits 15 to 2 = AD13 to AD0, bits 1 and 0 = 00b)
Fourth conversion: 0000h ≤ ADDRy (y = 0 to 4, 6, 8 to 15) ≤ FFF0h
(ADDRy (y = 0 to 4, 6, 8 to 15): Bits 15 to 2 = AD13 to AD0, bits 1 and 0 = 00b)

## 30.2.2 A/D Data Duplication Register (ADDBLDR)

ADDBLDR is a 16-bit read-only register used in double trigger mode. ADDBLDR holds the results of A/D conversion of the analog input of the channel selected for data duplication when the conversion is started by the second trigger. ADDBLDR uses the following different formats depending on the setting of the A/D data register format select bit (ADRFMT) in ADCER or A/D-converted value addition mode.

- ADCER.ADRFMT = 0 (Setting for right-alignment)

Address(es): 0008 9018h

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | AD[11:0] | | | | | | | | | | | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b11 to b0 | AD[11:0] | — | 12-bit A/D-converted value | R |
| b15 to b12 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |

- ADCER.ADRFMT = 1 (Setting for left-alignment)

Address(es): 0008 9018h

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | AD[11:0] | | | | | | | | | | | | — | — | — | — |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b3 to b0 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |
| b15 to b4 | AD[11:0] | — | 12-bit A/D-converted value | R |

- When A/D-converted value addition mode is selected

Address(es): 0008 9018h

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | AD[13:0] | | | | | | | | | | | | | | — | — |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b1, b0 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |
| b15 to b2 | AD[13:0] | — | 14-bit A/D-converted value addition result | R |

When A/D-converted value addition mode is selected, the ADDBLDR.AD[13:0] bits show the value added by the A/D-converted value of the respective channels. In A/D-converted value addition mode, the setting of the ADCER.ADRFMT bit in ADCER becomes invalid and the format of the register becomes left-aligned.

### 30.2.3 A/D Temperature Sensor Data Register (ADTSDR)

ADTSDR is a 16-bit read-only register that holds the A/D conversion results of the temperature sensor output. The following different formats are used depending on the settings of the A/D data register format select bit (ADCER.ADRFMT) and A/D-converted value addition mode.

- ADCER.ADRFMT = 0 (Setting for right-alignment)

Address(es): 0008 901Ah

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | AD[11:0] | | | | | | | | | | | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b11 to b0 | AD[11:0] | — | 12-bit A/D-converted value | R |
| b15 to b12 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |

- ADCER.ADRFMT = 1 (Setting for left-alignment)

Address(es): 0008 901Ah

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | AD[11:0] | | | | | | | | | | | | — | — | — | — |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b3 to b0 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |
| b15 to b4 | AD[11:0] | — | 12-bit A/D-converted value | R |

- When A/D-converted value addition function is selected

Address(es): 0008 901Ah

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | AD[13:0] | | | | | | | | | | | | | | — | — |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b1, b0 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |
| b15 to b2 | AD[13:0] | — | A/D-converted value addition result of temperature sensor output | R |

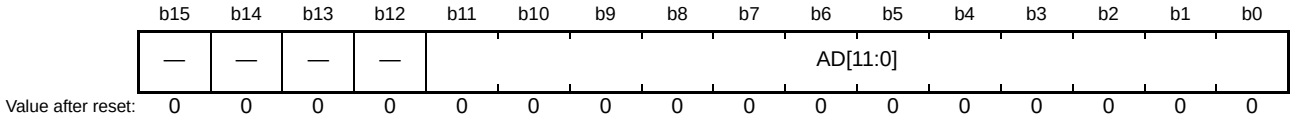
When A/D-converted value addition mode is selected, the ADDBLDR.AD[13:0] bits show the temperature sensor output value added by the A/D-converted value. In A/D-converted value addition mode, the setting of the ADCER.ADRFMT bit becomes invalid and the format of the register becomes left-aligned.

## 30.2.4 A/D Internal Reference Voltage Data Register (ADOCDR)

ADOCDR is a 16-bit read-only register that holds the A/D conversion results of the internal reference voltage.
The following different formats are used depending on the setting of the A/D data register format select bit (ADRFMT) in ADCER or A/D-converted value addition mode.

- ADCER.ADRFMT = 0 (Setting for right-alignment)

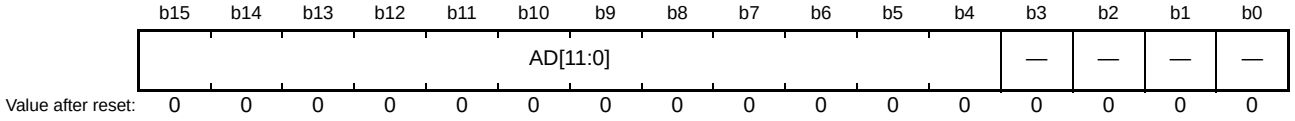
Address(es): 0008 901Ch




| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b11 to b0 | AD[11:0] | — | 12-bit A/D-converted value | R |
| b15 to b12 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |

- ADCER.ADRFMT = 1 (Setting for left-alignment)

Address(es): 0008 901Ch




| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b3 to b0 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |
| b15 to b4 | AD[11:0] | — | 12-bit A/D-converted value | R |

- When A/D-converted value addition mode is selected

Address(es): 0008 901Ch


| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b1, b0 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |
| b15 to b2 | AD[13:0] | — | 14-bit A/D-converted value addition result | R |

When A/D-converted value addition mode is selected, the AD[13:0] bits in ADOCDR show the value added by the A/D-converted value of the internal reference voltage. In A/D-converted value addition mode, the setting of the ADRFMT bit in ADCER becomes invalid and the format of the register becomes left-aligned.

## 30.2.5 A/D Control Register (ADCSR)

Address(es): 0008 9000h

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | ADST | ADCS[1:0] | | ADIE | — | ADHSC | TRGE | EXTRG | DBLE | GBADIE | — | DBLANS[4:0] | | | | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b4 to b0 | DBLANS[4:0] | A/D Conversion Data Duplication Channel Select | Select one of 14 analog input channels for A/D conversion data duplication. These bits are valid only in double trigger mode. | R/W |
| b5 | — | Reserved | This bit is read as 0. The write value should be 0. | R/W |
| b6 | GBADIE | Group B Scan End Interrupt Enable | 0: Disables GBADI interrupt generation upon group B scan completion.<br>1: Enables GBADI interrupt generation upon group B scan completion. | R/W |
| b7 | DBLE | Double Trigger Mode Select | 0: Deselects double trigger mode.<br>1: Selects double trigger mode. | R/W |
| b8 | EXTRG | Trigger Select*1 | 0: A/D conversion is started by the synchronous trigger (MTU or ELC).<br>1: A/D conversion is started by the asynchronous trigger (ADTRG0#). | R/W |
| b9 | TRGE | Trigger Start Enable | 0: Disables A/D conversion to be started by the synchronous or asynchronous trigger.<br>1: Enables A/D conversion to be started by the synchronous or asynchronous trigger. | R/W |
| b10 | ADHSC | A/D Conversion Mode Select | 0: Normal conversion<br>1: High-speed conversion | R/W |
| b11 | — | Reserved | This bit is read as 0. The write value should be 0. | R/W |
| b12 | ADIE | Scan End Interrupt Enable | 0: Disables S12ADI0 interrupt generation upon scan completion.<br>1: Enables S12ADI0 interrupt generation upon scan completion. | R/W |
| b14, b13 | ADCS[1:0] | Scan Mode Select | b14 b13<br>0 0: Single scan mode<br>0 1: Group scan mode<br>1 0: Continuous scan mode<br>1 1: Setting prohibited | R/W |
| b15 | ADST | A/D Conversion Start | 0: Stops A/D conversion.<br>1: Starts A/D conversion. | R/W |

Note 1. Starting A/D conversion using an external pin (asynchronous trigger)
If 1 is written to both the TRGE and EXTRG bits in ADCSR when a high-level signal is input to the external pin (ADTRG0#), and then if the ADTRG0# signal is driven low, the falling edge of ADTRG0# is detected and the scan conversion process is started. In this case, the pulse width of the low-level input must be at least 1.5 PCLK clock cycles.

### DBLANS[4:0] Bits (A/D Conversion Data Duplication Channel Select)

The DBLANS[4:0] bits select one of the channels for A/D conversion data duplication in double trigger mode. The A/D conversion results of the analog input of the selected channel are stored into A/D data register y when conversion is started by the first trigger, and into the A/D data duplication register when started by the second trigger. Table 30.4 shows the relationship between the DBLANS[4:0] bit settings and selected duplication channel. A/D-converted value addition mode with double trigger mode can be set by selecting the channel selected by the DBLANS[4:0] bits using the ADADS register. If double trigger mode is selected, the channel selected by the ADANSA register is invalid, and the channel selected by the DBLANS[4:0] bits is subjected to A/D conversion instead. When converting analog inputs of channels, temperature sensor output and internal reference voltage should not be selected for A/D conversion.
The DBLANS[4:0] bits should be set while the ADST bit is 0 (they should not be set simultaneously when 1 is written to the ADST bit.)

**Table 30.4 Relationship between DBLANS[4:0] Bit Settings and Double Trigger Enabled Channels**

| DBLANS[4:0] | Duplication Channel | DBLANS[4:0] | Duplication Channel |
|---|---|---|---|
| 00000 | AN000 | 01001 | AN009 |
| 00001 | AN001 | 01010 | AN010 |
| 00010 | AN002 | 01011 | AN011 |
| 00011 | AN003 | 01100 | AN012 |
| 00100 | AN004 | 01101 | AN013 |
| 00110 | AN006 | 01110 | AN014 |
| 01000 | AN008 | 01111 | AN015 |

### GBADIE Bit (Group B Scan End Interrupt Enable)

The GBADIE bit enables or disables group B scan end interrupt (GBADI) in group scan mode.

### DBLE Bit (Double Trigger Mode Select)

In double trigger mode, the following operation is performed after scanning is started by the MTU or ELC trigger selected by the TRSA[3:0] bits in ADSTRGR.

1. When the ADIE bit is 1, a scan end interrupt is not output upon first scan completion but is output upon second scan completion.
2. The A/D conversion results of the analog input of the channel selected by the DBLANS[4:0] bits are stored into A/D data register y for the first time, and into the A/D data duplication register for the second time.

Setting the DBLE bit to 1 invalidates the channel selected by the ADANSA register. In continuous scan mode, double trigger mode should not be selected. Temperature sensor output and internal reference voltage should not be selected for A/D conversion. In double trigger mode, software trigger should not be selected. The DBLE bit should be set while ADST bit is 0 (it should not be set simultaneously when 1 is written to the ADST bit.)

### TRGE Bit (Trigger Start Enable)

The TRGE bit enables or disables A/D conversion by the synchronous trigger and the asynchronous trigger.
This bit should be set to 1 in group scan mode.

### ADHSC Bit (A/D Conversion Mode Select)

This bit selects whether A/D conversion mode is normal conversion mode or high-speed conversion mode.
Normal conversion mode can be selected when AVCC0 is in the range of 1.8 V to 2.4 V.
High-speed conversion mode can be selected when AVCC0 ≥ 2.4 V.
Conversion can be performed in 1 μs if the ADCLK clock is 32 MHz when AVCC0 ≥ 2.7 V and high-speed conversion mode is selected.

### ADIE Bit (Scan End Interrupt Enable)

The ADIE bit enables or disables the A/D scan end interrupt (S12ADI0) in scans except for group B scan in group scan mode.
With double trigger mode deselected, the S12ADI0 interrupt is generated when the first scan is completed if the ADIE bit is set to 1.
With temperature sensor output or internal reference voltage being selected, the S12ADI0 interrupt is also generated when A/D conversion is completed if the ADIE bit is set to 1.
With double trigger mode selected, the S12ADI0 interrupt is generated when the second scan is completed if the ADIE bit is set to 1 as long as the scan is started by the MTU or ELC trigger selected by the ADSTRGR.TRSA[3:0] bits.
When scanning is started by a software trigger, even with double trigger mode selected, the S12ADI0 interrupt is generated when scanning is completed if the ADIE bit is set to 1.

### ADCS[1:0] Bits (Scan Mode Select)

The ADCS[1:0] bits select the scan mode.
In single scan mode, A/D conversion is performed for the analog inputs of a maximum of 14 channels selected with the ADANSA register in the ascending order of the channel number, and when one cycle of A/D conversion is completed for all the selected channels, scan conversion is stopped.
In continuous scan mode, while the ADCSR.ADST bit is 1, A/D conversion is performed for the analog inputs of a maximum of 14 channels selected with the ADANSA register in the ascending order of the channel number, and when one cycle of A/D conversion is completed for all the selected channels, A/D conversion is repeated beginning at the first channel. A/D conversion is repeated until the ADCSR.ADST bit is set to 0.
In group scan mode, A/D conversion is performed for the analog inputs (group A) of a maximum of 14 channels selected with the ADANSA register in the ascending order of the channel number after scanning is started by the MTU or ELC trigger selected by the ADSTRGR.TRSA[3:0] bits, and when one cycle of A/D conversion is completed for all the selected channels, A/D conversion is stopped. A/D conversion is also performed for the analog inputs (group B) of a maximum of 14 channels selected with the ADANSB register in the ascending order of the channel number after scanning is started by the MTU or ELC trigger selected by the ADSTRGR.TRSA[3:0] bits, and when one cycle of A/D conversion is completed for all the selected channels, A/D conversion is stopped. In group scan mode, different channels and triggers should be selected for group A and group B.
When temperature sensor output or internal reference voltage is selected, single scan mode should be selected and all the channels selected by the ADANSA register should be deselected, after which A/D conversion is to be started. A/D conversion stops after completion of A/D conversion of the temperature sensor output or the internal reference voltage selected.
The ADCS[1:0] bits should be set while the ADST bit is 0 (it should not be set simultaneously when 1 is written to the ADST bit.)

### ADST Bit (A/D Conversion Start)

The ADST bit starts or stops A/D conversion process.
Before the ADST bit is set to 1, set the A/D conversion clock, the conversion mode, and conversion target analog input.
[Setting conditions]

- 1 is written by software.
- The synchronous trigger (MTU or ELC) selected by the ADSTRGR.TRSA[3:0] bits is detected with ADCSR.EXTRG and ADCSR.TRGE bits being set to 0 and 1, respectively.
- A synchronous trigger (MTU or ELC) selected by the ADSTRGR.TRSB[3:0] bits is detected with the ADCSR.TRGE bit being set to 1 in group scan mode.
- The asynchronous trigger is detected with the ADCSR.TRGE and ADCSR.EXTRG bits being set to 1 and the ADSTRGR.TRSA[3:0] bits being set to 0000b.

[Clearing conditions]

- 0 is written by software.
- The A/D conversion of all the channels selected is completed in single scan mode.
- The A/D conversion of the temperature sensor output or the internal reference voltage selected is completed in single scan mode.
- Group A scan is completed in group scan mode.
- Group B scan is completed in group scan mode.

## 30.2.6 A/D Channel Select Register A (ADANSA)

Address(es): 0008 9004h

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | ANSA[15:8] | | | | | | | | — | ANSA[6] | — | ANSA[4:0] | | | | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b4 to b0 | ANSA[4:0] | A/D Conversion Channel 0 to 4 Select | 0: AN000 to AN004 are not subjected to conversion.<br>1: AN000 to AN004 are subjected to scan conversion. | R/W |
| b5 | — | Reserved | This bit is read as 0. The write value should be 0. | R/W |
| b6 | ANSA[6] | A/D Conversion Channel 6 Select | 0: AN006 is not subjected to conversion.<br>1: AN006 is subjected to scan conversion. | R/W |
| b7 | — | Reserved | This bit is read as 0. The write value should be 0. | R/W |
| b15 to b8 | ANSA[15:8] | A/D Conversion Channel 8 to 15 Select | 0: AN008 to AN015 are not subjected to conversion.<br>1: AN008 to AN015 are subjected to scan conversion. | R/W |

The ADANSA register selects analog input channels for A/D conversion from among AN000 to AN004, AN006, AN008 to AN015. In group scan mode, group A channels are to be selected.

**ANSA[15:8, 6, 4:0] Bits (A/D Conversion Channels 0 to 4, 6, 8 to 15 Select)**

The ANSA[15:8, 6, 4:0] bits select analog input channels for A/D conversion from among AN000 to AN004, AN006, AN008 to AN015. The channels to be selected and the number of channels can be arbitrarily set. The ANSA[0] bit corresponds to AN000 and the ANSA[15] bit corresponds to AN015. When A/D conversion of analog inputs of the channels is to be performed, A/D conversion of the temperature sensor output and the internal reference voltage should not be performed.

When double trigger mode is selected, the channel selected by the ANSA[15:8, 6, 4:0] bits is invalid, and the channel selected by the ADCSR.DBLANS[4:0] bits is selected in group A instead.

The ANSA[15:8, 6, 4:0] bits should be set while the ADCSR.ADST bit is 0.

## 30.2.7 A/D Channel Select Register B (ADANSB)

Address(es): 0008 9014h

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | ANSB[15:8] | | | | | | | | — | ANSB[6] | — | ANSB[4:0] | | | | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b4 to b0 | ANSB[4:0] | A/D Conversion Channels 0 to 4 Select | 0: AN000 to AN004 are not subjected to conversion.<br>1: AN000 to AN004 are subjected to scan conversion. | R/W |
| b5 | — | Reserved | This bit is read as 0. The write value should be 0. | R/W |
| b6 | ANSB[6] | A/D Conversion Channel 6 Select | 0: AN006 is not subjected to conversion.<br>1: AN006 is subjected to scan conversion. | R/W |
| b7 | — | Reserved | This bit is read as 0. The write value should be 0. | R/W |
| b15 to b8 | ANSB[15:8] | A/D Conversion Channels 8 to 15 Select | 0: AN008 to AN015 are not subjected to conversion.<br>1: AN008 to AN015 are subjected to scan conversion. | R/W |

The ADANSB register selects analog inputs 0 to 4, 6, and 8 to 15 of the channels for A/D conversion in group B in group scan mode. The ADANSB register is not used in any other scan mode.

**ANSB[15:8, 6, 4:0] Bits (A/D Conversion Channels 0 to 4, 6, 8 to 15 Select)**

The ADANSB register selects channels for A/D conversion in group B from among AN000 to AN004, AN006, AN008 to AN015 in group scan mode. ADANSB is not used in any other scan mode. The channels for conversion can be selected from among the channels other than group A channels, which are selected by the ADANSA register or ADCSR.DBLANS[4:0] bits in double trigger mode. The ANSB[0] bit corresponds to AN000 and the ANSB[15] bit corresponds to AN015. When A/D conversion of analog inputs of the channels is to be performed, A/D conversion of the temperature sensor output and the internal reference voltage should not be performed.

The ANSB[15:8, 6, 4:0] bits should be set while the ADST bit is 0.

## 30.2.8 A/D-Converted Value Addition Mode Select Register (ADADS)

Address(es): 0008 9008h

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | ADS[15:8] | | | | | | | | — | ADS[6] | — | ADS[4:0] | | | | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b4 to b0 | ADS[4:0] | A/D-Converted Value Addition Channels 4 to 0 Select | 0: A/D-converted value addition function for AN004 to AN000 are not selected.<br>1: A/D-converted value addition function for AN004 to AN000 are selected. | R/W |
| b5 | — | Reserved | This bit is read as 0. The write value should be 0. | R/W |
| b6 | ADS[6] | A/D-Converted Value Addition Channel 6 Select | 0: A/D-converted value addition function for AN006 is not selected.<br>1: A/D-converted value addition function for AN006 is selected. | R/W |
| b7 | — | Reserved | This bit is read as 0. The write value should be 0. | R/W |
| b15 to b8 | ADS[15:8] | A/D-Converted Value Addition Channels 15 to 8 Select | 0: A/D-converted value addition function for AN015 to AN008 are not selected.<br>1: A/D-converted value addition function for AN015 to AN008 are selected. | R/W |

The ADADS register selects the channels 0 to 4, 6, 8 to 15 on which A/D conversion is performed successively two to four times and then converted values are added (integrated).

**ADS[15:8, 6, 4:0] Bits (A/D-Converted Value Addition Channels 15 to 8, 6, 4 to 0 Select)**

When the ADS[n] bit of the number that is the same as that of A/D converted channel selected by the ADANSA.ANSA[n] bits (n = 0 to 4, 6, 8 to 15) or ADCSR.DBLANS[4:0] bits and ADANSB.ANSB[n] bits (n = 0 to 4, 6, 8 to 15) is set to 1, these bits perform A/D conversion of analog input of the selected channels successively two to four times that is set with the ADADC.ADC[1:0] bits and returns the added (integrated) conversion results to the A/D data register. For the channel for which the A/D conversion is performed and addition mode is not selected, a normal one-time conversion is performed and the conversion result is returned to the A/D data register.
The ADS[15:8, 6, 4:0] bits should be set while the ADCSR.ADST bit is 0.

Figure 30.2 shows a scanning operation sequence in which both the ADS[2] and ADS[4] bits are set to 1.
In continuous scan mode (ADCSR.ADCS[1:0] = 10b), it is assumed that the addition count is set to 3 (ADADC.ADC[1:0] = 11b) and the channels AN000 to AN004, AN006 are selected (ADANSA.ANSA[15:0] = 005Fh). The conversion process begins with AN000. The AN002 conversion is performed successively 4 times, and the added (integrated) value is returned to A/D data register 2. After that the AN003 conversion process is started. The AN004 conversion is performed successively 4 times and the added (integrated) value is returned to A/D data register 4. After conversion of AN006, the conversion operation is once again performed in the same sequence from AN000.
For the channel for which the addition mode is not selected, the A/D data register format is determined by the ADCER.ADRFMT bit (right-alignment or left-alignment).

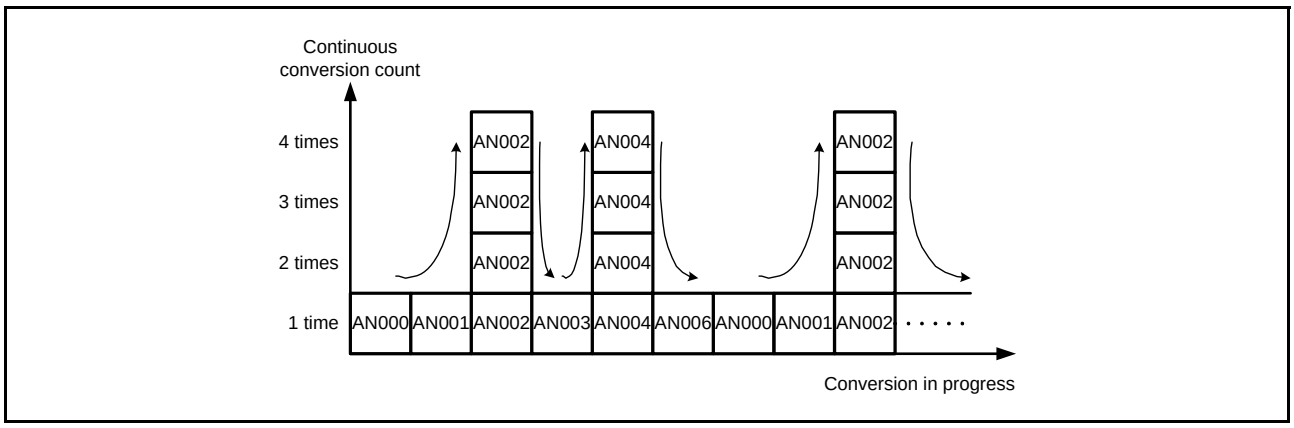



**Figure 30.2 Scan Conversion Sequence with ADADC.ADC[1:0] = 11b, ADS[2] = 1, and ADS[4] = 1**

## 30.2.9 A/D-Converted Value Addition Count Select Register (ADADC)

Address(es): 0008 900Ch

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | ADC[1:0] | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b1, b0 | ADC[1:0] | Addition Count Select | b1 b0<br>0 0: 1-time conversion (no addition; same as normal conversion)<br>0 1: 2-time conversion (addition once)<br>1 0: 3-time conversion (addition twice)<br>1 1: 4-time conversion (addition three times) | R/W |
| b7 to b2 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |

The ADADC register sets the addition count for the channels for which A/D-converted value addition mode is selected, and for A/D conversion of the temperature sensor output and the internal reference voltage.

**ADC[1:0] Bits (Addition Count Select)**

These bits set the addition count common to the channels for which A/D conversion or A/D-converted value addition mode is selected, including the channels selected in double trigger mode (by the ADCSR.DBLANS[4:0] bits), and to A/D conversion of the temperature sensor output and the internal reference voltage.
The ADC[1:0] bits should be set while the ADCSR.ADST bit is 0.

## 30.2.10 A/D Control Extended Register (ADCER)

Address(es): 0008 900Eh

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | ADRFMT | — | — | — | — | — | — | — | — | — | ACE | — | — | — | — | — |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b4 to b0 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |
| b5 | ACE | Automatic Clearing Enable | 0: Disables automatic clearing.<br>1: Enables automatic clearing. | R/W |
| b14 to b6 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |
| b15 | ADRFMT | A/D Data Register Format Select | 0: Right-alignment is selected for the A/D data register format.<br>1: Left-alignment is selected for the A/D data register format. | R/W |

### ACE Bit (Automatic Clearing Enable)

The ACE bit enables or disables automatic clearing (All 0) of ADDRy, ADOCDR, ADDBLDR, and ADTSDR after the register has been read by the CPU or DTC. This function enables update failures of the ADDRy, ADOCDR, ADDBLDR, and ADTSDR registers to be detected.

### ADRFMT Bit (A/D Data Register Format Select)

The ADRFMT bit specifies left-alignment or right-alignment for the data to be stored in ADDRy, ADOCDR, ADDBLDR, and ADTSDR.

When the A/D-converted value addition mode is selected, the format of each data register is fixed to left-alignment, irrespective of the ADCER.ADRFMT bit value.

For details on the format of the data registers, refer to section 30.2.1, A/D Data Registers y (ADDRy) (y = 0 to 4, 6, 8 to 15), section 30.2.2, A/D Data Duplication Register (ADDBLDR), section 30.2.3, A/D Temperature Sensor Data Register (ADTSDR), and section 30.2.4, A/D Internal Reference Voltage Data Register (ADOCDR).

## 30.2.11 A/D Start Trigger Select Register (ADSTRGR)

Address(es): 0008 9010h

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | TRSA[3:0] | | | | — | — | — | — | TRSB[3:0] | | | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b3 to b0 | TRSB[3:0] | A/D Conversion Start Trigger Select for Group B | Select the A/D conversion start trigger for group B in group scan mode. | R/W |
| b7 to b4 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |
| b11 to b8 | TRSA[3:0] | A/D Conversion Start Trigger Select | Select the A/D conversion start trigger in single scan mode and continuous mode. In group scan mode, the A/D conversion start trigger for group A is selected. | R/W |
| b15 to b12 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |

### TRSB[3:0] Bits (A/D Conversion Start Trigger Select for Group B)

The TRSB[3:0] bits select the trigger to start scanning of the analog input selected in group B. The TRSB[3:0] bits require to be set only in group scan mode and are not used in any other scan mode. For the scan conversion start trigger for group B, setting software trigger or asynchronous trigger is prohibited. Therefore, the TRSB[3:0] bits should be set to the value other than 0000 and the ADCSR.TRGE bit should be set to 1 in group scan mode.

Table 30.5 lists the A/D conversion startup sources selected by the TRSB[3:0] bits.

### TRSA[3:0] Bits (A/D Conversion Start Trigger Select)

The TRSA[3:0] bits select the trigger to start A/D conversion in single scan mode and continuous scan mode. In group scan mode, the trigger to start scanning of the analog input selected in group A is selected. For scan execution in group scan mode or double trigger mode, software trigger or asynchronous trigger cannot be used.

- When using the A/D conversion startup source of the synchronous trigger (MTU and ELC), set the ADCSR.TRGE bit to 1 and set the ADCSR.EXTRG bit to 0.
- When using the asynchronous trigger (ADTRG0#), set the ADCSR.TRGE bit to 1 and set the ADCSR.EXTRG bit to 1.
- Software trigger (ADCSR.ADST) is enabled regardless of the set values of the ADCSR.TRGE and ADCSR.EXTRG bits and the TRSA[3:0] bits.

Table 30.6 lists the A/D conversion startup sources selected by the TRSA[3:0] bits.

**Table 30.5 List of A/D Conversion Startup Sources Selected by TRSB[3:0] Bits**

| Module | Source | Remarks | TRSB[3] | TRSB[2] | TRSB[1] | TRSB[0] |
|---|---|---|---|---|---|---|
| MTU | TRG0AN | TRGA input capture/compare match from MTU0 | 0 | 0 | 0 | 1 |
| | TRG0BN | TRGB input capture/compare match B from MTU0 | 0 | 0 | 1 | 0 |
| | TRGAN | TRGA input capture/compare match or MTU4.TCNT underflow (trough) in complementary PWM mode from MTU0 to MTU4 | 0 | 0 | 1 | 1 |
| | TRG0EN | TRGE compare match from MTU0 | 0 | 1 | 0 | 0 |
| | TRG0FN | TRGF compare match from MTU0 | 0 | 1 | 0 | 1 |
| | TRG4AN | MTU4.TADCORA and MTU4.TCNT compare match (interrupt skipping function 1) | 0 | 1 | 1 | 0 |
| | TRG4BN | MTU4.TADCORB and MTU4.TCNT compare match (interrupt skipping function 1) | 0 | 1 | 1 | 1 |
| | TRG4ABN | MTU4.TADCORA and MTU4.TCNT compare match and MTU4.TADCORB and MTU4.TCNT compare match (interrupt skipping function 1) | 1 | 0 | 0 | 0 |
| ELC | ELC | Trigger from ELC | 1 | 0 | 0 | 1 |

**Table 30.6 List of A/D Conversion Startup Sources Selected by TRSA[3:0] Bits**

| Module | Source | Remarks | TRSA[3] | TRSA[2] | TRSA[1] | TRSA[0] |
|---|---|---|---|---|---|---|
| ADC | ADST | Software trigger | — | — | — | — |
| External input | ADTRG0# | A/D conversion start trigger pin | 0 | 0 | 0 | 0 |
| MTU | TRG0AN | TRGA input capture/compare match from MTU0 | 0 | 0 | 0 | 1 |
| | TRG0BN | TRGB input capture/compare match B from MTU0 | 0 | 0 | 1 | 0 |
| | TRGAN | TRGA input capture/compare match or MTU4.TCNT underflow (trough) in complementary PWM mode from MTU0 to MTU4 | 0 | 0 | 1 | 1 |
| | TRG0EN | TRGE compare match from MTU0 | 0 | 1 | 0 | 0 |
| | TRG0FN | TRGF compare match from MTU0 | 0 | 1 | 0 | 1 |
| | TRG4AN | MTU4.TADCORA and MTU4.TCNT compare match (interrupt skipping function 1) | 0 | 1 | 1 | 0 |
| | TRG4BN | MTU4.TADCORB and MTU4.TCNT compare match (interrupt skipping function 1) | 0 | 1 | 1 | 1 |
| | TRG4ABN | MTU4.TADCORA and MTU4.TCNT compare match and MTU4.TADCORB and MTU4.TCNT compare match (interrupt skipping function 1) | 1 | 0 | 0 | 0 |
| ELC | ELC | Trigger from ELC | 1 | 0 | 0 | 1 |

## 30.2.12 A/D Converted Extended Input Control Register (ADEXICR)

Address(es): 0008 9012h

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | OCS | TSS | — | — | — | — | — | — | OCSAD | TSSAD |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b0 | TSSAD | Temperature Sensor Output A/D-Converted Value Addition Function Select | 0: Temperature sensor output A/D-converted value addition function is not selected.<br>1: Temperature sensor output A/D-converted value addition function is selected. | R/W |
| b1 | OCSAD | Internal Reference Voltage A/D-Converted Value Addition Mode Select | 0: Internal reference voltage A/D-converted value addition mode is not selected<br>1: Internal reference voltage A/D-converted value addition mode is selected | R/W |
| b7 to b2 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |
| b8 | TSS | Temperature Sensor Output A/D Conversion Select | 0: A/D conversion of temperature sensor output is not performed<br>1: A/D conversion of temperature sensor output is performed | R/W |
| b9 | OCS | Internal Reference Voltage A/D Conversion Select | 0: A/D conversion of internal reference voltage is not performed<br>1: A/D conversion of internal reference voltage is performed | R/W |
| b15 to b10 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |

### TSSAD Bit (Temperature Sensor Output A/D-Converted Value Addition Function Select)

When the TSSAD bit is set to 1, A/D conversion of temperature sensor output is selected and performed successively 2 to 4 times that is set with the ADADC.ADC[1:0] bits, and the added value is returned to the A/D temperature sensor data register (ADTSDR). The TSSAD bit should be set while the ADCSR.ADST bit is 0.

### OCSAD Bit (Internal Reference Voltage A/D-Converted Value Addition Mode Select)

The OCSAD bit selects A/D conversion for the internal reference voltage. Setting the OCSAD bit to 1 performs A/D conversion of the internal reference voltage successively two to four times that is set with the ADADC.ADC[1:0] bits and returns the integrated value to the A/D internal reference voltage data register (ADOCDR). The OCSAD bit should be set while the ADCSR.ADST bit is 0.

### TSS Bit (Temperature Sensor Output A/D Conversion Select)

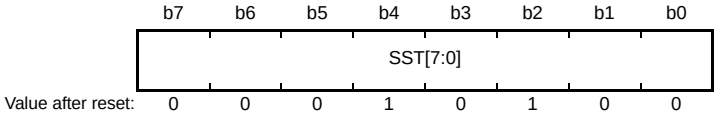
The TSS bit selects A/D conversion for the temperature sensor output. When A/D conversion of the temperature sensor output is to be performed, all the bits in registers ADANSA and ADANSB, the ADCER.DIAGM bit, the ADCSR.DBLE bit, and the OCS bit should be set to 0 in single scan mode. The TSS bit should be set while the ADST bit is 0.
Do not use the first conversion result after the TSS bit is set to 1. Insert a stabilization wait time of 5 µs after the first conversion is completed and before the second conversion is started.

### OCS Bit (Internal Reference Voltage A/D Conversion Select)

The OCS bit selects A/D conversion for the internal reference voltage. When A/D conversion of the internal reference voltage is to be performed, all the bits in the ADANSA register should be set to 0 in single scan mode. The OCS bit should be set while the ADST bit is 0.
Do not use the first conversion result after the OCS bit is set to 1. Insert a stabilization wait time of 5 µs after the first conversion is completed and before the second conversion is started.

## 30.2.13 A/D Sampling State Register n (ADSSTRn) (n = 0 to 4, 6, L, T, O)

Address(es): ADSSTR0: 0008 9060h, ADSSTR1: 0008 9073h, ADSSTR2: 0008 9074h, ADSSTR3: 0008 9075h, ADSSTR4: 0008 9076h, ADSSTR6: 0008 9078h, ADSSTRL: 0008 9061h, ADSSTRT: 0008 9070h, ADSSTRO: 0008 9071h




| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b7 to b0 | SST[7:0] | Sampling Time LO Setting | Sets the sampling time for each state (6 to 255 states). | R/W |

ADSSTRn sets the sampling time for analog input.
The actual sampling time is the register setting value + one state.
One state is one ADCLK (A/D conversion clock) cycle. When the ADCLK is 32 MHz, one state is 31.25 ns. The initial value is 20 states. If the impedance of the analog input signal source is too high to secure sufficient sampling time or if the ADCLK is slow, the sampling time can be adjusted. These bits should be set while the ADCSR.ADST bit is 0. The set value for sampling time should be between 6 to 255 states.
Table 30.7 shows the A/D sampling state registers and corresponding channels.

**Table 30.7 A/D Sampling State Registers and Corresponding Channels**

| Bit Name | Corresponding Channels |
|---|---|
| ADSSTR0.SST[7:0] | AN000 |
| ADSSTR1.SST[7:0] | AN001 |
| ADSSTR2.SST[7:0] | AN002 |
| ADSSTR3.SST[7:0] | AN003 |
| ADSSTR4.SST[7:0] | AN004 |
| ADSSTR6.SST[7:0] | AN006 |
| ADSSTRL.SST[7:0] | AN008 to AN015 |
| ADSSTRT.SST[7:0] | Temperature sensor output |
| ADSSTRO.SST[7:0] | Internal reference voltage |

## 30.3 Operation

### 30.3.1 Scanning Operation

In scanning, A/D conversion is performed sequentially on the analog inputs of the specified channels.
A scan conversion is performed in three operating modes: single scan mode, continuous scan mode, and group scan mode. Also, conversion modes are divided into high-speed conversion mode and normal conversion mode. In single scan mode, one or more specified channels are scanned once. In continuous scan mode, one or more specified channels are scanned repeatedly until the ADCSR.ADST bit is cleared to 0 from 1 by software. In group scan mode, the selected channels of group A and the selected channels of group B are scanned once after starting to be scanned according to the respective triggers.
In single scan mode and continuous scan mode, A/D conversion is performed for ANn channels selected by the ADANSA register, starting from the channel with the smallest number n. In group scan mode, A/D conversion is performed for ANn channels of group A and group B selected by the ADANSA and ADANSB registers, respectively, starting from the channel with the smallest number n.
High-speed mode can be selected when AVCC0 ≥ 2.4 V. Normal conversion mode can be selected when AVCC0 ≥ 1.8 V. High-speed mode reduces conversion time by six cycles compared to normal conversion mode.
When temperature sensor output or internal reference voltage is selected, single scan mode should be used for A/D conversion.
This operation is similar to the scan operation when only one channel is selected in single scan mode.
Double trigger mode is to be used with single scan mode or group scan mode. With double trigger mode being enabled, A/D conversion data of a channel selected by the ADCSR.DBLANS[4:0] bits is duplicated only if the conversion is started by any of the MTU or ELC triggers selected by the ADSTRGR.TRSA[3:0] bits.
In any scanning mode, the software trigger, synchronous trigger, or asynchronous trigger input to be used as the A/D conversion start condition is disabled while the ADCSR.ADST bit is 1 (during scanning process).

## 30.3.2 Single Scan Mode

### 30.3.2.1 Basic Operation

In basic operation of single scan mode, A/D conversion is performed once on the analog input of the specified channels as below. In selected channel scanning, the temperature sensor output A/D conversion select bit (ADEXICR.TSS) and the internal reference voltage A/D conversion select bit (ADEXICR.OCS) should be set to 0 (not selected). The following describes an operation when double trigger mode is not selected.

(1) When the ADCSR.ADST bit is set to 1 (A/D conversion start) by software, synchronous trigger (MTU or ELC), or asynchronous trigger input, A/D conversion is performed for ANn channels selected by the ADANSA register, starting from the channel with the smallest number n.

(2) Each time A/D conversion of a single channel is completed, the A/D conversion result is stored into the corresponding A/D data register (ADDRy).

(3) When A/D conversion of all the selected channels is completed, an S12ADI0 interrupt request is generated if the ADCSR.ADIE bit is 1 (S12ADI0 interrupt upon scanning completion enabled).

(4) The ADST bit remains 1 (A/D conversion start) during A/D conversion, and is automatically cleared to 0 when A/D conversion of all the selected channels is completed. Then the 12-bit A/D converter enters a wait state.

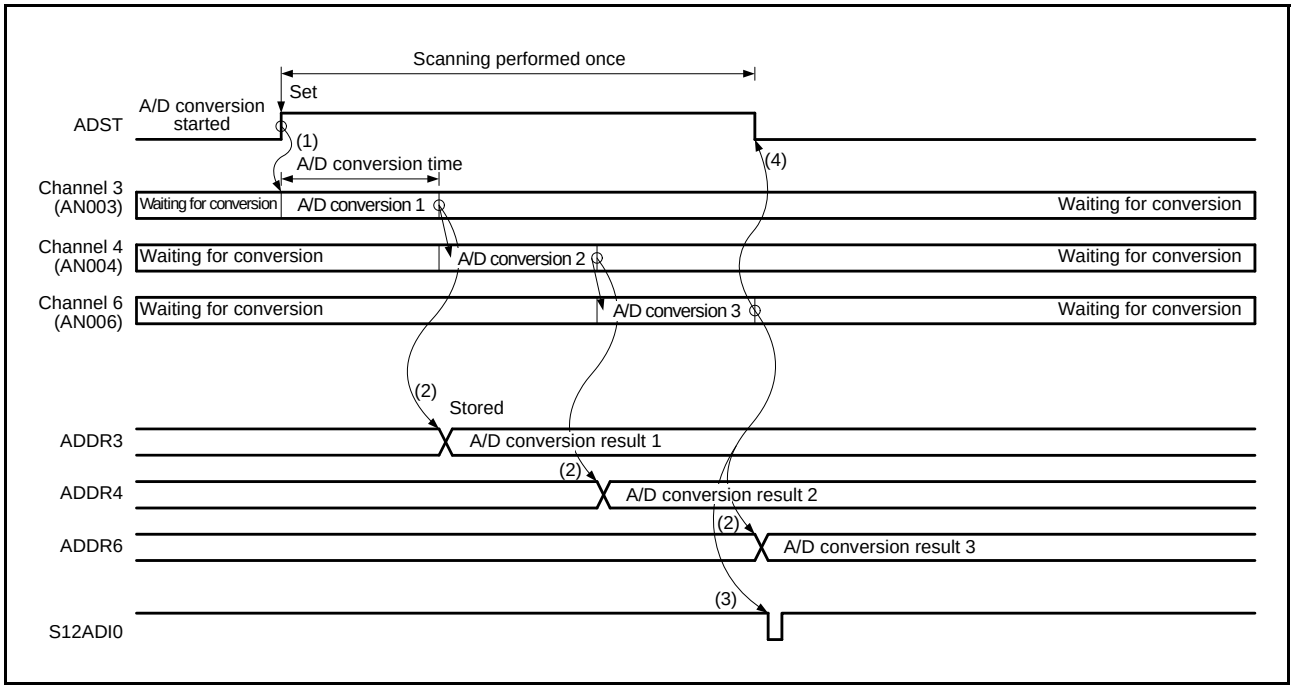



**Figure 30.3 Example of Operation in Single Scan Mode (Basic Operation: AN003, AN004, AN006 Selected)**

### 30.3.2.2 A/D Conversion when Temperature Sensor Output is Selected

To perform A/D conversion of the temperature sensor output, make the following settings before starting conversion operation.

- Set the ADCSR.ADST bit to 0 (stops A/D conversion).
- Set the ADCSR.ADCS[1:0] bits to 00b (single scan mode).
- Set the ADANSA register to 0000h (all external analog inputs are not subject to conversion)
- Set the ADCSR.DBLE bit to 0 (deselects double trigger mode).
- Set the ADEXICR.TSS bit to 1 (A/D conversion of temperature sensor output is performed).
- Set the ADEXICR.OCS bit to 0 (A/D conversion of internal reference voltage is not performed).

Set an appropriate value in the ADSSTRO register to set the sampling time to 5 µs or longer.

Figure 30.4 shows the procedure for A/D conversion of the temperature sensor output.

(1) Set the ADST bit to 1 (starts A/D conversion) to discharge the electric charge stored in the A/D converter ((1) in the figure).
Do not use the conversion result at this time.

(2) When a trigger is input or the ADST bit is set to 1, A/D conversion is started for the temperature sensor output ((2) in the figure).

(3) When A/D conversion is completed, the conversion result is stored into the ADOCDR register. If the ADCSR.ADIE bit is 1 (enables S12ADI0 interrupt generation upon scan completion.) at this time, an S12ADI0 interrupt request is generated ((3) in the figure).

(4) The ADST bit is changed to 0 and the A/D converter enters waits a wait state ((4) in the figure).

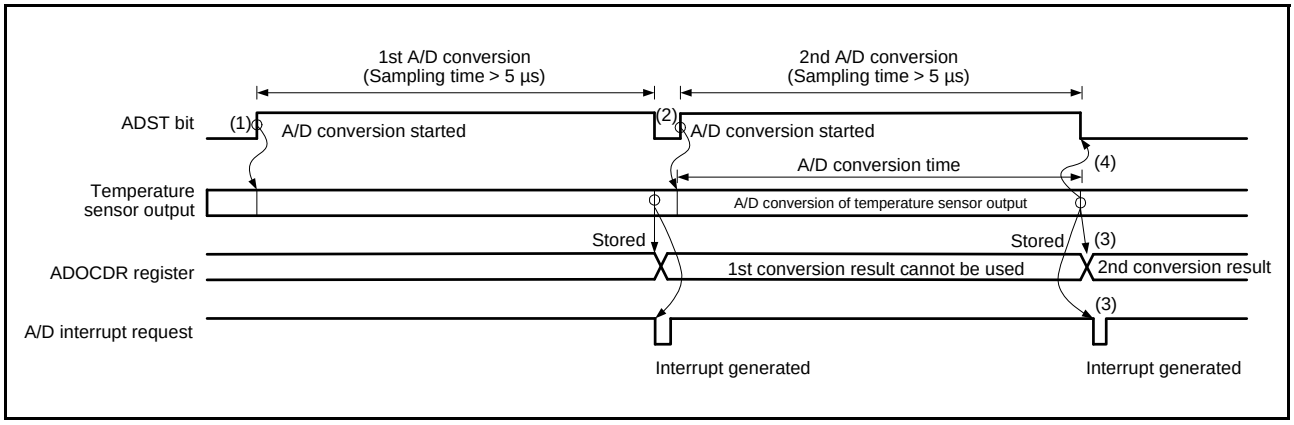



**Figure 30.4 Example of Operation in Single Scan Mode (Temperature Sensor Output Selected)**

### 30.3.2.3 A/D Conversion when Internal Reference Voltage is Selected

To perform A/D conversion of the internal reference voltage, make the following settings before starting conversion operation.

- Set the ADCSR.ADST bit to 0 (stops A/D conversion).
- Set the ADCSR.ADCS[1:0] bits to 00b (single scan mode).
- Set the ADANSA register to 0000h (all external analog inputs are not subject to conversion)
- Set the ADCSR.DBLE bit to 0 (deselects double trigger mode).
- Set the ADEXICR.TSS bit to 0 (A/D conversion of temperature sensor output is not performed).
- Set the ADEXICR.OCS bit to 1 (A/D conversion of internal reference voltage is performed).

Set an appropriate value in the ADSSTRO register to set the sampling time to 5 µs or longer.

Figure 30.5 shows the procedure for A/D conversion of the internal reference voltage.

(1) Set the ADST bit to 1 (starts A/D conversion) to discharge the electric charge stored in the A/D converter ((1) in the figure).
Do not use the conversion result at this time.

(2) When a trigger is input or the ADST bit is set to 1, A/D conversion is started for the internal reference voltage ((2) in the figure).

(3) When A/D conversion is completed, the conversion result is stored into the ADOCDR register. If the ADCSR.ADIE bit is 1 (enables S12ADI0 interrupt generation upon scan completion.) at this time, an S12ADI0 interrupt request is generated ((3) in the figure).

(4) The ADST bit is changed to 0 and the A/D converter enters waits a wait state ((4) in the figure).

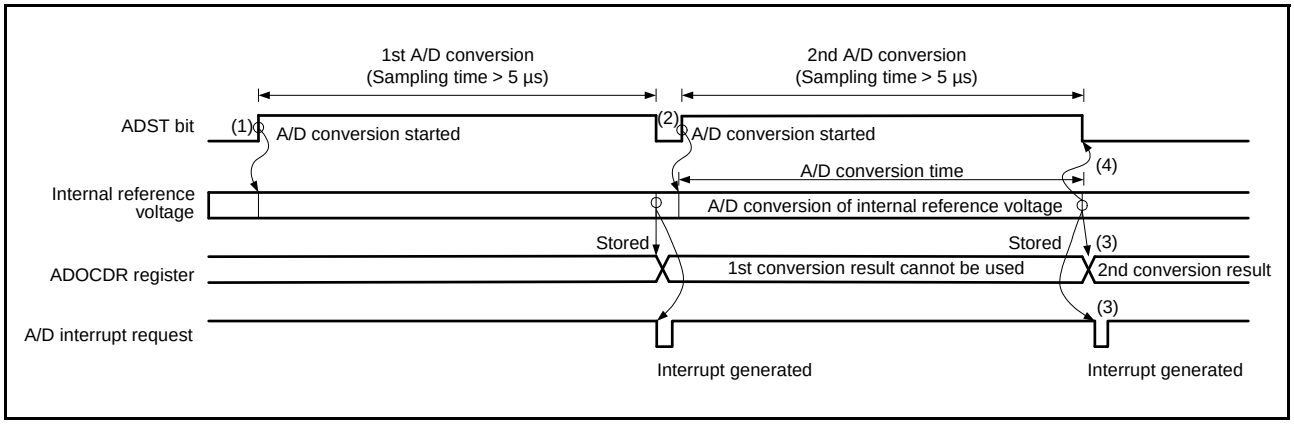



**Figure 30.5 Example of Operation in Single Scan Mode (Internal Reference Voltage Selected)**

### 30.3.2.4 A/D Conversion in Double Trigger Mode

In single scan mode with double trigger mode, single scan operation started by synchronous trigger is performed twice as below.

The temperature sensor output A/D conversion select bit (ADEXICR.TSS) and the internal reference voltage A/D conversion select bit (ADEXICR.OCS) should be set to 0 (not selected).

Duplication of A/D conversion data is enabled by setting the channel numbers to be duplicated to the DBLANS[4:0] bits in ADCSR and setting the DBLE bit in ADCSR to 1. When the DBLE bit in ADCSR is set to 1, channel selection using the ADANSA register is invalid. In double trigger mode, synchronous triggers should be selected using the ADSTRGR.TRSA[3:0] bits, the ADCSR.EXTRG bit should be set to 0, and the ADCSR.TRGE bit should be set to 1. Software trigger should not be used.

(1) When the ADCSR.ADST bit is set to 1 (A/D conversion start) by synchronous trigger input, A/D conversion is started on the single channel selected by the ADCSR.DBLANS[4:0] bits.

(2) When A/D conversion is completed, the A/D conversion result is stored into the corresponding A/D data register (ADDRy).

(3) The ADST bit is automatically cleared to 0 and the 12-bit A/D converter enters a wait state. Here, an S12ADI0 interrupt request is not generated irrespective of the ADIE (S12ADI0 interrupt upon scanning completion enabled) bit setting in ADCSR.

(4) When the ADST bit in ADCSR is set to 1 (A/D conversion start) by the second trigger input, A/D conversion is started on the single channel selected by the DBLANS[4:0] bits in ADCSR.

(5) When A/D conversion is completed, the A/D conversion result is stored into the A/D data duplication register (ADDBLDR), which is exclusively used in double trigger mode.

(6) If the ADIE bit in ADCSR is 1 (S12ADI0 interrupt upon scanning completion enabled), an S12ADI0 interrupt request is generated.

(7) The ADST bit remains 1 (A/D conversion start) during A/D conversion, and is automatically cleared to 0 when A/D conversion is completed. Then the 12-bit A/D converter enters a wait state.

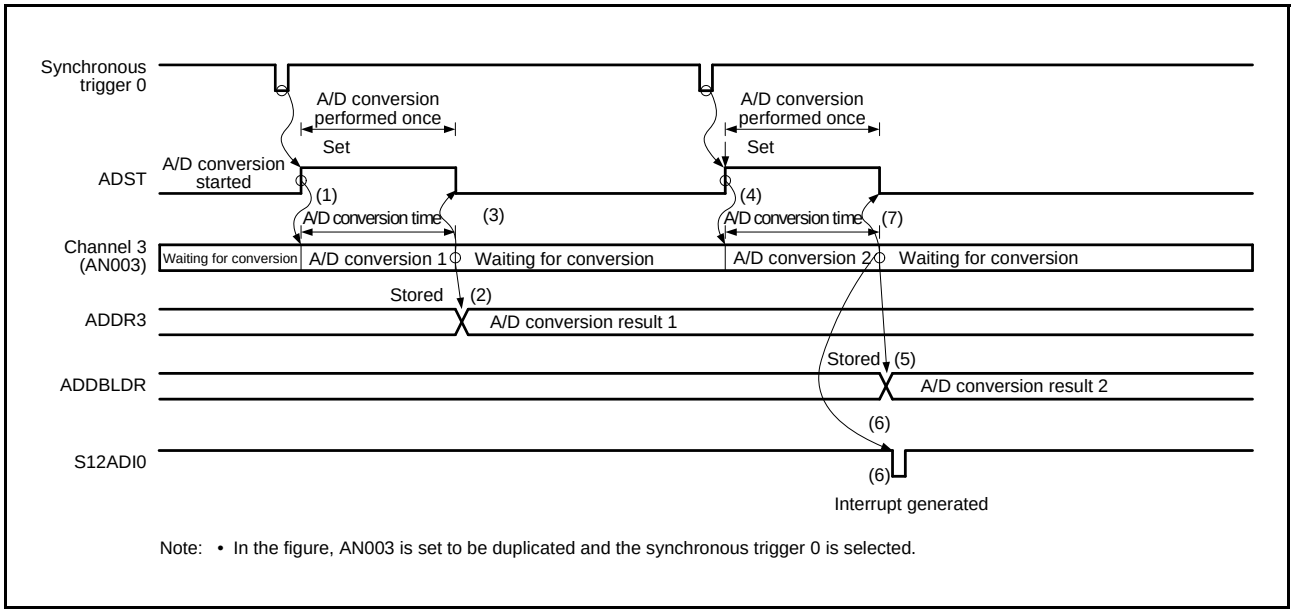



**Figure 30.6 Example of Operation in Single Scan Mode (Double Trigger Mode Selected; AN003 Duplicated)**

## 30.3.3 Continuous Scan Mode

### 30.3.3.1 Basic Operation

In basic operation of continuous scan mode, A/D conversion is performed repeatedly on the analog input of the specified channels as below.

In continuous scan mode, the temperature sensor output A/D conversion select bit (ADEXICR.TSS) and the internal reference voltage A/D conversion select bit (ADEXICR.OCS) should both be set to 0 (not selected).

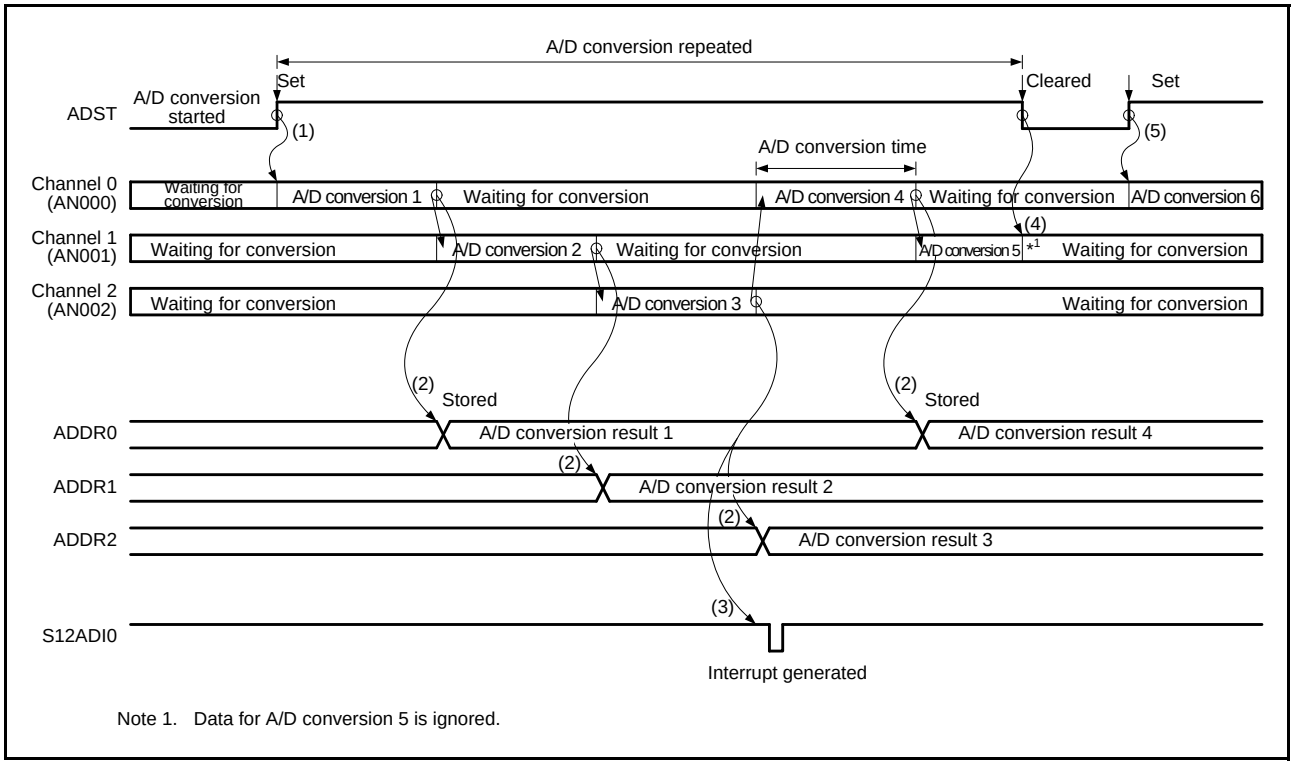
(1) When the ADST bit in ADCSR is set to 1 (A/D conversion start) by software, synchronous trigger (MTU or ELC), or asynchronous trigger input, A/D conversion is performed for ANn channels selected by the ADANSA register, starting from the channel with the smallest number n.

(2) Each time A/D conversion of a single channel is completed, the A/D conversion result is stored into the corresponding A/D data register (ADDRy).

(3) When A/D conversion of all the selected channels is completed, an S12ADI0 interrupt request is generated if the ADIE bit in ADCSR is 1 (S12ADI0 interrupt upon scanning completion enabled).
The 12-bit A/D converter sequentially starts A/D conversion for ANn channels selected by the ADANSA register, starting from the channel with the smallest number n.

(4) The ADST bit in ADCSR is not automatically cleared to 0 and steps 2 and 3 are repeated as long as the bit remains 1 (A/D conversion start). When the ADST bit in ADCSR is set to 0 (A/D conversion stop), A/D conversion stops and the 12-bit A/D converter enters a wait state.

(5) When the ADST bit is later set to 1 (A/D conversion start), A/D conversion is started again for ANn channels selected by the ADANSA register, starting from the channel with the smallest number n.




Note 1. Data for A/D conversion 5 is ignored.

**Figure 30.7 Example of Operation in Continuous Scan Mode (Basic Operation: AN000 to AN002 Selected)**

## 30.3.4 Group Scan Mode

### 30.3.4.1 Basic Operation

In basic operation of group scan mode, A/D conversion is performed once on the analog inputs of all the specified channels in group A and group B after scanning is started by synchronous trigger as below. Scan operation of each group is similar to the scan operation in single scan mode.

The group A trigger and group B trigger can be selected using the TRSA[3:0] and TRSB[3:0] bits in ADSTRGR, respectively. The different triggers should be used for group A and group B to prevent simultaneous A/D conversion of group A and group B. Software trigger should not be used.

The group A and group B channels to be A/D-converted are selected using the ADANSA register and ADANSB register, respectively. Group A and group B cannot use the same channels.

In group scan mode, the temperature sensor output A/D conversion select bit (ADEXICR.TSS) and the internal reference voltage A/D conversion select bit (ADEXICR.OCS) should both be set to 0 (not selected).

The following describes operation in group scan mode using a trigger from the MTU. Specifically, the TRG4AN and TRG4BN triggers from the MTU are assumed to be used to start conversion of group A and group B, respectively.

(1) Scanning of group A is started by the TRG4AN trigger from the MTU.
(2) When group A scanning is completed, an S12ADI0 interrupt is output if the ADIE bit in ADCSR is 1 (S12ADI0 interrupt enabled).
(3) Scanning of group B is started by the TRG4BN trigger from the MTU.
(4) When group B scanning is completed, a GBADI interrupt is output if the GBADIE bit in ADCSR is 1 (GBADI interrupt enabled).

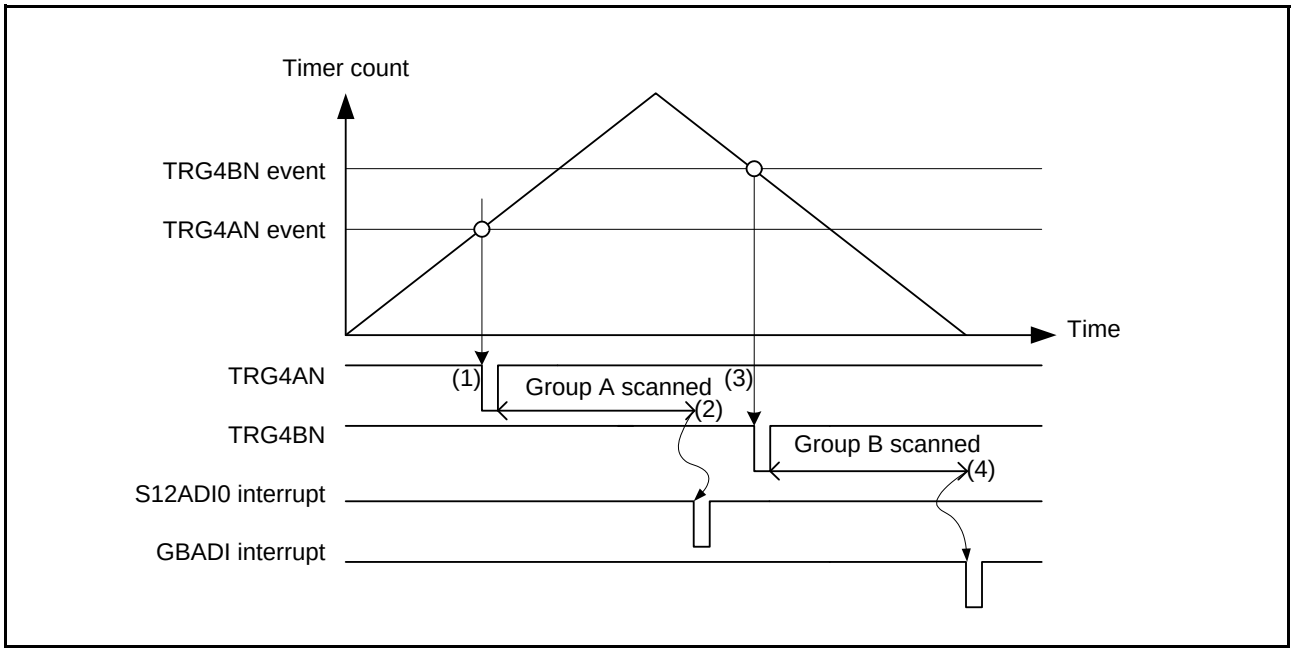



**Figure 30.8 Example of Operation in Group Scan Mode (Basic Operation: MTU Triggers Used)**

### 30.3.4.2 A/D Conversion in Double Trigger Mode

In group scan mode with double trigger mode, single scan operation started by synchronous trigger is performed twice for group A. For group B, single scan operation started by synchronous trigger is performed once.
In group scan mode, the group A trigger and group B trigger can be selected using the TRSA[3:0] and TRSB[3:0] bits in ADSTRGR, respectively. The different triggers should be used for group A and group B to prevent simultaneous A/D conversion of group A and group B. Software trigger, or asynchronous trigger (ADTRG0#) should not be used.
The group A and group B channels to be A/D-converted are selected using the DBLANS[4:0] bits in the ADCSR register and ADANSB register, respectively. The same channels cannot be selected for both groups.
In group scan mode, the temperature sensor output A/D conversion select bit (ADEXICR.TSS) and the internal reference voltage A/D conversion select bit (ADEXICR.OCS) should both be set to 0 (not selected).
Duplication of A/D conversion data is enabled by setting the channel numbers to be duplicated to the DBLANS[4:0] bits in ADCSR and setting the DBLE bit in ADCSR to 1.
The following describes operation in group scan mode with double trigger mode using a trigger from the MTU. Specifically, the TRG4BN and TRG0AN triggers from the MTU are assumed to be used to start conversion of group A and group B, respectively.

(1) Scanning of group B is started by the TRG0AN trigger from the MTU.
(2) When group B scanning is completed, a GBADI interrupt is output if the GBADIE bit in ADCSR is 1 (GBADI interrupt enabled).
(3) The first scanning of group A is started by the first TRG4BN trigger from the MTU.
(4) When the first scanning of group A is completed, the conversion result is stored into ADDRy; an S12ADI0 interrupt request is not generated irrespective of the ADIE bit setting in ADCSR.
(5) The second scanning of group A is started by the second TRG4BN trigger from the MTU.
(6) When the second scanning of group A is completed, the conversion result is stored into ADDBLDR. An S12ADI0 interrupt is output if the ADIE bit is 1 (S12ADI0 interrupt enabled).

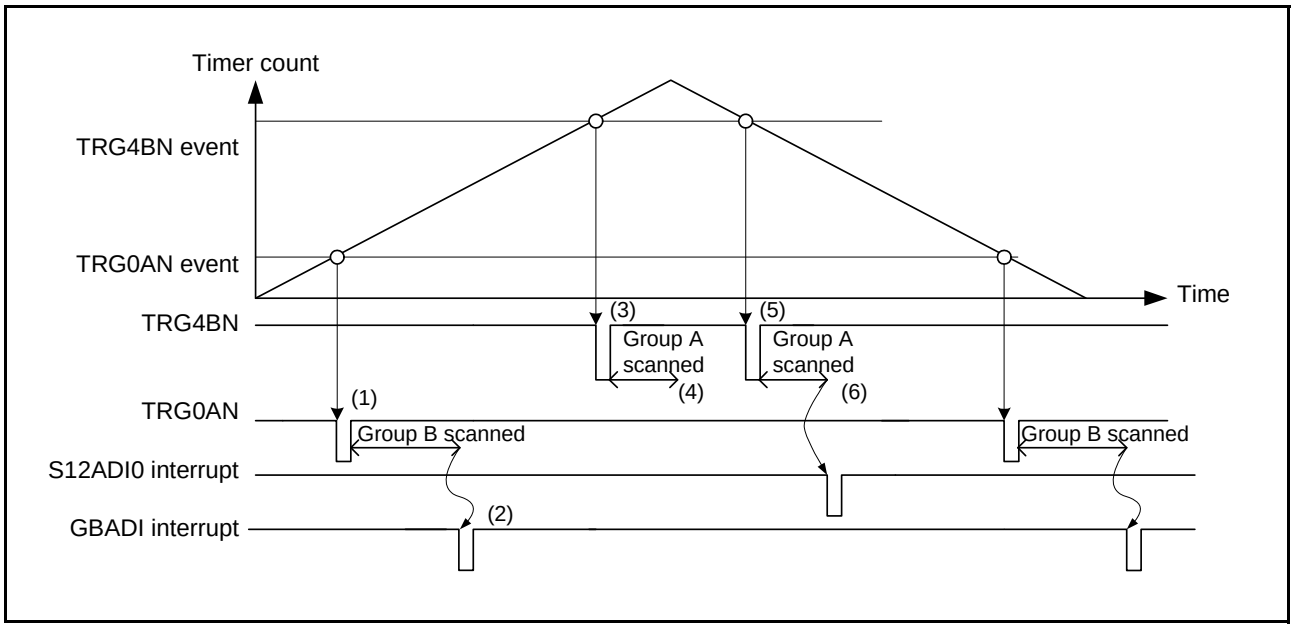



**Figure 30.9 Example of Operation in Group Scan Mode with Double Trigger Mode (Basic Operation: MTU Triggers Used)**

### 30.3.4.3 Notes on Using Software Trigger

When a software trigger is input with double trigger mode selected, scanning of the selected channels is performed, and the S12ADI0 interrupt is output if the ADCSR.ADIE bit is 1 (enables S12ADI0 interrupt), regardless of scanning even or odd number of times. In addition, data is not duplicated even if scanning by a software trigger is performed even number of times.

The following shows an example when a software trigger is input while scanning by a synchronous trigger with double trigger mode selected.

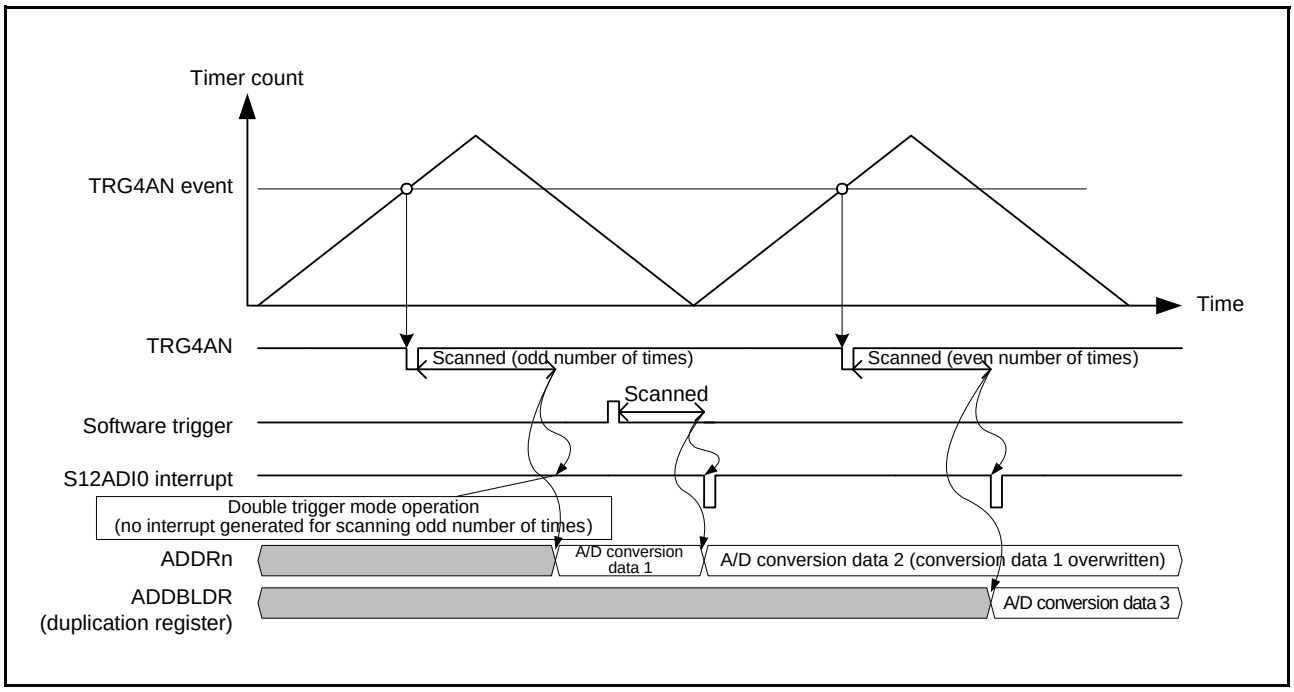



**Figure 30.10 Example of Double Trigger Operation When Using Software Trigger**

### 30.3.5 Analog Input Sampling and Scan Conversion Time

Scan conversion can be activated either by software trigger; the triggers from the MTU, ELC, or ADTRG0# (external trigger). After start-of-scanning-delay time ($t_D$) has passed, then starts the A/D conversion process.
Figure 30.11 shows the scan conversion timing in single scan mode, in which scan conversion is activated by software trigger or triggers from the MTU or ELC. Figure 30.12 shows the scan conversion timing in single scan mode, in which scan conversion is activated by ADTRG0# (external trigger). The scan conversion time ($t_{SCAN}$) includes start-of-scanning-delay time ($t_D$), A/D conversion processing time ($t_{CONV}$), and end-of-scanning-delay time ($t_{ED}$). Table 30.8 shows the specific scanning time.
The scan conversion time ($t_{SCAN}$) in single scan mode for which the number of selected channels is n can be determined as follows:

$$t_{SCAN} = t_D + (t_{SPL} + t_{CONV}) \, n + t_{ED}$$

The scan conversion time for the first cycle in continuous scan mode is $t_{SCAN}$ for single scan minus $t_{ED}$.
The scan conversion time for the second and subsequent cycles in continuous scan mode is fixed to $(t_{SPL} + t_{CONV}) \, n$.

**Table 30.8 Scan Conversion Time (in Terms of PCLK and ADCLK Cycles)**

| Item | Symbol | Conditions | Scan Conversion Time (Cycles) |
|---|---|---|---|
| Start-of-scanning-delay time*1 | $t_D$ | MTU, ELC, or software trigger | 2 PCLK + 3 ADCLK |
| | | External trigger | 4 PCLK + 3 ADCLK |
| Sampling time*1 | $t_{SPL}$ | ADSSTRn.SST[7:0] bits (initial set value 14h) | (register set value + 1) ADCLK |
| A/D conversion processing time*1 | $t_{CONV}$ | High-speed mode | 23 ADCLK |
| | | Normal-speed mode | 29 ADCLK |
| Scan conversion time*1 | $t_{ED}$ | — | 1 PCLK + 2 ADCLK*2 |

Note 1. For $t_D$, $t_{SPL}$, $t_{CONV}$, and $t_{ED}$, see Figure 30.11 and Figure 30.12.
Note 2. The value of 2 ADCLK is fixed and an interrupt is output within plus 1 PCLK. For details on the processing time for termination, refer to section 30.7.3, A/D Conversion Restarting Timing and Termination Timing.

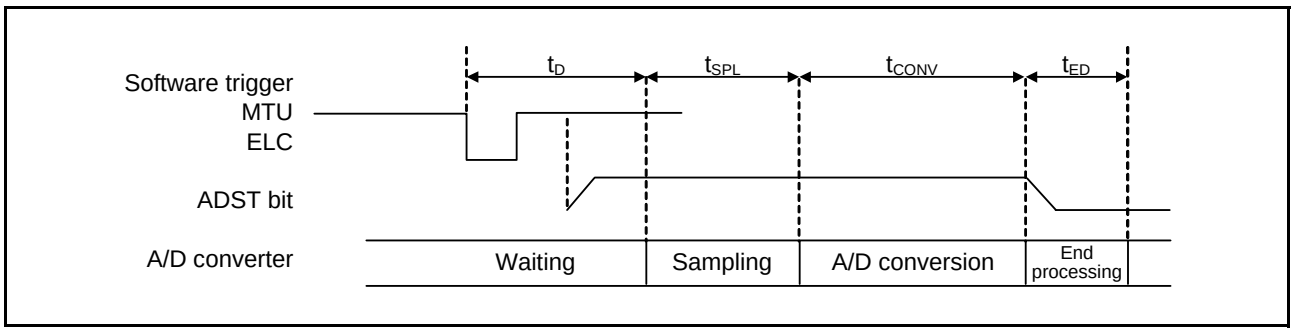



**Figure 30.11 Scan Conversion Timing (Activated by Software, or Triggers from the MTU or ELC)**

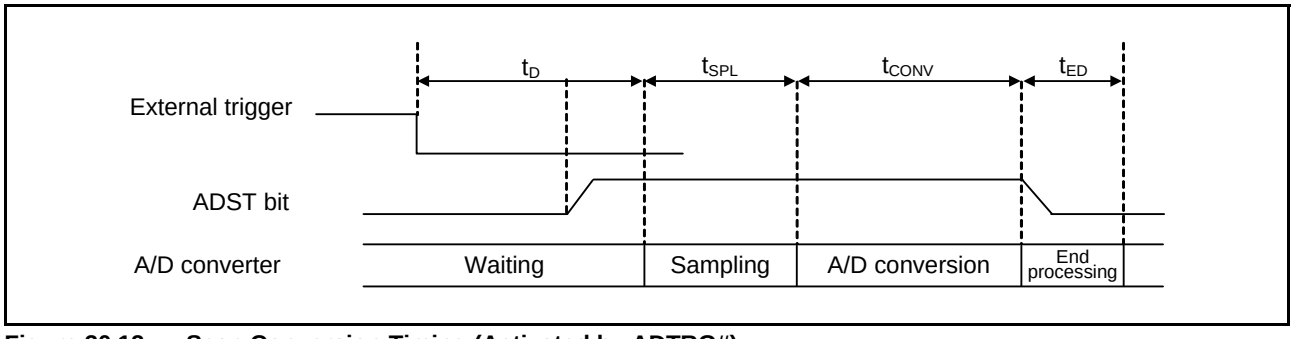



**Figure 30.12 Scan Conversion Timing (Activated by ADTRG#)**

### 30.3.6 Usage Example of Automatic Register Clearing Function

Setting the ADCER.ACE bit to 1 automatically clears the A/D data registers (ADDRy, ADOCDR, ADTSDR, and ADDBLDR) to 0000h when the A/D data registers are read by the CPU or DTC.
This function enables detection of update failures of the A/D data registers. The following describes the examples in which the function to automatically clear the ADDRy register is enabled and disabled.
In a case where the ADCER.ACE bit is 0 (automatic clearing is disabled), if the A/D conversion result (0222h) is not written to the ADDRy register for some reason, the old data (0111h) will be the ADDRy value. Furthermore, if this ADDRy value is written to a general register using an A/D scan end interrupt, the old data (0111h) can be saved in the general register. When checking whether there is an update failure, it is necessary to frequently save the old data in the RAM or a general register.
In a case where the ADCER.ACE bit is 1 (automatic clearing is enabled), when ADDRy = 0111h is read by the CPU, DTC, or ADDRy is automatically cleared to 0000h. After that, if the A/D conversion result 0222h cannot be transferred to ADDRy for some reason, the cleared data (0000h) remains as the ADDRy value. If this ADDRy value is read into a general register using an A/D scan end interrupt at this point, 0000h will be saved in the general register. Occurrence of an ADDRy update failure can be determined by simply checking that the read data value is 0000h.

### 30.3.7 A/D-Converted Value Addition Function

The same channel is A/D converted two to four consecutive times and the sum of the converted values is stored in the data register. The use of the average of these results can improve the accuracy of A/D conversion, depending on the types of noise components that are present. This function, however, cannot always guarantee an improvement in A/D conversion accuracy.
A/D converted value addition function can be used at the time of selecting the channel selection analog input A/D conversion, temperature sensor output A/D conversion, or A/D internal reference voltage A/D conversion.

### 30.3.8 Starting A/D Conversion with an Asynchronous Trigger

The A/D conversion can be started by the input of an asynchronous trigger. To start up the A/D converter by an asynchronous trigger, the A/D conversion start trigger select bits (ADSTRGR.TRSA[3:0]) should be set to 0000b and a high-level signal should be input to the asynchronous trigger (ADTRG0# pin), and both the ADCSR.TRGE and ADCSR.EXTRG bits should be set to 1. Figure 30.13 shows a timing of the asynchronous trigger input.
For the time required for the A/D conversion start after the ADCSR.ADST bit is set, refer to section 30.7.3, A/D Conversion Restarting Timing and Termination Timing.
An asynchronous trigger cannot be selected by the A/D conversion start trigger select bits (ADSTRGR.TRSA[3:0]) for group B to be used in group scan mode.

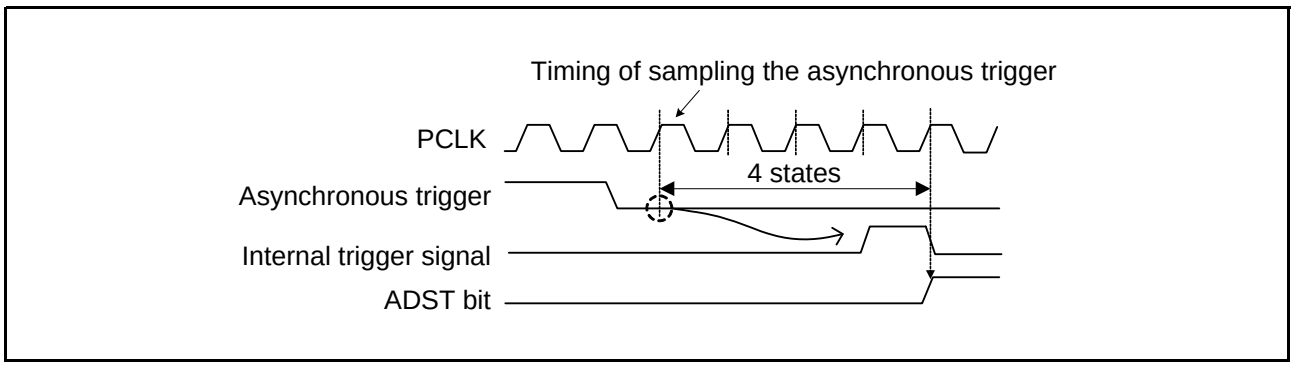



**Figure 30.13 Asynchronous Trigger Input Timing**

### 30.3.9 Starting A/D Conversion with Synchronous Trigger from Peripheral Modules

The A/D conversion can be started by a synchronous trigger of the MTU or ELC. To start the A/D conversion by a synchronous trigger, the ADCSR.TRGE bit should be set to 1, the ADCSR.EXTRG bit should be cleared to 0, and the relevant source should be selected by ADSTRGR.TRSA[3:0] and TRSB[3:0] bits.
The A/D conversion startup sources for group B to be used in group scan mode should be selected by the ADSTRGR.TRSB[3:0] bits. In group scan mode, the different A/D start conversion startup sources should be selected by the ADSTRGR.TRSA[3:0] bits and the ADSTRGR.TRSB[3:0] bits.

## 30.4 Interrupt Sources

### 30.4.1 Interrupt Request on Completion of Each Scanning Conversion

The 12-bit A/D converter can send scan end interrupt requests S12ADI0 and GBADI to the CPU.
Setting the ADCSR.ADIE bit to 1 and 0 enables and disables an S12ADI0 interrupt, respectively. Similarly, setting the ADCSR.GBADIE bit to 1 and 0 enables and disables a GBADI interrupt, respectively.
In addition, the DTC can be started up when an S12ADI0 or a GBADI interrupt is generated. Using an S12ADI0 or a GBADI interrupt to allow the DTC to read the converted data enables continuous conversion without burden on software. For details on DTC settings, refer to section 16, Data Transfer Controller (DTCa).

## 30.5 Event Linkage

### 30.5.1 Event Output to ELC

The ELC connects the S12ADI0 interrupt request signal to the predetermined module as the event signal (i.e., event linkage). The GBADI interrupt request signal cannot be used as the event signal. The event signal can be output irrespective of the setting of the corresponding interrupt request enable bit. The 12-bit A/D converter outputs the A/D conversion end event.

### 30.5.2 12-bit A/D Converter Operation by Event from ELC

The 12-bit A/D converter can be started by the predetermined event by so setting ELSRn of the ELC.

### 30.5.3 Notes on Event Reception from ELC during 12-bit A/D Conversion

When an event occurs during A/D conversion, it is invalid.

## 30.6 A/D Conversion Accuracy Definitions

The MCU's A/D conversion accuracy is defined as below:

- Resolution
  The number of 12-bit A/D converter digital output codes
- Offset error
  The deviation of the analog input voltage value from the ideal A/D conversion characteristic when the digital output changes from the minimum voltage value 000000000000 to 000000000001, excluding quantization error.
- Full-scale error
  The deviation of the analog input voltage value from the ideal A/D conversion characteristic when the digital output changes from 111111111110 to 111111111111, excluding quantization error.
- Quantization error
  The deviation inherent in the 12-bit A/D converter, given by 1/2 LSB
- Nonlinearity error
  The error with respect to the ideal A/D conversion characteristic between zero voltage and the full-scale error, excluding offset error, full-scale error, and quantization error.
- Absolute accuracy
  The deviation between the digital value and analog input value, including offset error, full-scale error, quantization error, and nonlinearity error.

## 30.7 Usage Notes

### 30.7.1 Notes on Reading Data Registers

The A/D data registers, A/D data duplication register, A/D temperature sensor data register, and A/D internal reference voltage data register should be read in word units. If a register is read twice in byte units, that is, the higher-order byte and lower-order byte are separately read, the A/D converted value having been read first may disagree with the A/D converted value having been read for the second time. To prevent this, the data registers should never be read in byte units.

### 30.7.2 Notes on Stopping A/D Conversion

To stop A/D conversion when an asynchronous trigger or a synchronous trigger has been selected as the condition for starting A/D conversion, set the ADCSR.TRGE bit to 0 and select the software trigger as the condition for starting A/D conversion, and then set the ADCSR.TRGE bit to 0 (to stop A/D conversion).

### 30.7.3 A/D Conversion Restarting Timing and Termination Timing

It takes a maximum of four ADCLK cycles for the idle analog unit of the 12-bit A/D converter to be restarted by setting the ADCSR.TRGE bit to 1. It takes a maximum of two ADCLK cycles for the operating analog unit of the 12-bit A/D converter to be terminated by setting the ADCSR.TRGE bit to 0.

### 30.7.4 Notes on Scan End Interrupt Handling

When scanning the same analog input twice using any trigger, the first A/D converted data is overwritten with the second A/D converted data in the case that the CPU does not complete reading the A/D converted data by the time the A/D conversion of the first analog input for the second scan ends after the first scan end interrupt is generated.

### 30.7.5 Module Stop Function Setting

Operation of the 12-bit A/D converter can be disabled or enabled by setting module stop control register A (MSTPCRA). The initial setting is for operation of the 12-bit A/D converter to be halted. Register access is enabled by clearing the module stop state.
After the module stop state is canceled, wait for 1 µs to start A/D conversion. For details, refer to section 11, Low Power Consumption.

### 30.7.6 Notes on Entering Low Power Consumption States

Before entering module stop mode or software standby mode, make sure to stop A/D conversion. Here, set the ADCSR.ADST bit to 0, and allow time for stopping the analog unit of the 12-bit A/D converter.
Follow the procedure given below to secure this time.
1. Set the ADCSR.TRGE bit to 0 (software trigger).
2. Clear the ADCSR.ADST bit to 0.
3. After confirming that the A/D converter has been stopped, place the MCU in the module stop state mode or software standby mode.

### 30.7.7 Notes on Canceling Software Standby Mode

After software standby mode is canceled, wait until the crystal oscillation stabilization time or the PLL circuit stabilization time elapses, and then wait for 1 µs before starting A/D conversion. For details, refer to section 11, Low Power Consumption.

### 30.7.8 Allowable Impedance of Signal Source

To achieve high-speed conversion of 1.0 µs, the analog input pins of this MCU are designed so that the conversion accuracy is guaranteed if the impedance of the input signal source is 0.3 kΩ or less. If an external capacitor of large capacitance is attached in the application in which only a single pin input is converted in single scan mode, the only load on input is virtually 2.6 kΩ of the internal input resistor; therefore, the impedance of the signal source can be ignored. Being a low-pass filter, however, an analog input circuit may not follow the analog signal with a large differential coefficient as shown in Figure 30.14. When high-speed analog signals are to be converted or multiple pins are to be converted in scan mode, a low-impedance buffer should be used.

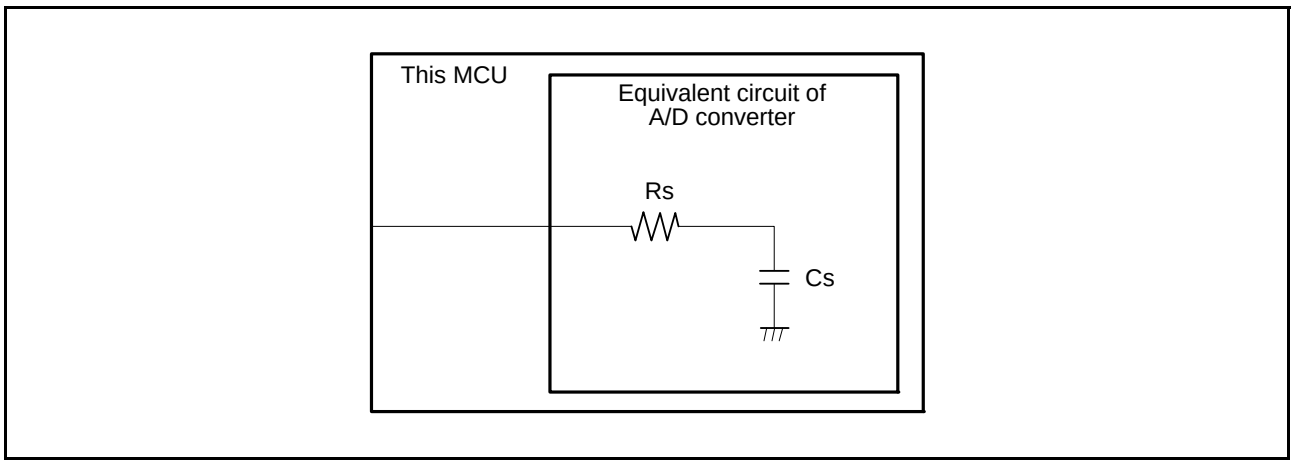



**Figure 30.14 Internal Equivalent Circuit of Analog Input Pin**

Figure 30.15 shows an equivalent circuit of an analog input pin and an external sensor.
To perform A/D conversion accurately, charging of the internal capacitor C shown in Figure 30.15 must be completed within the specified period of time. This specified period is referred to as sampling time.

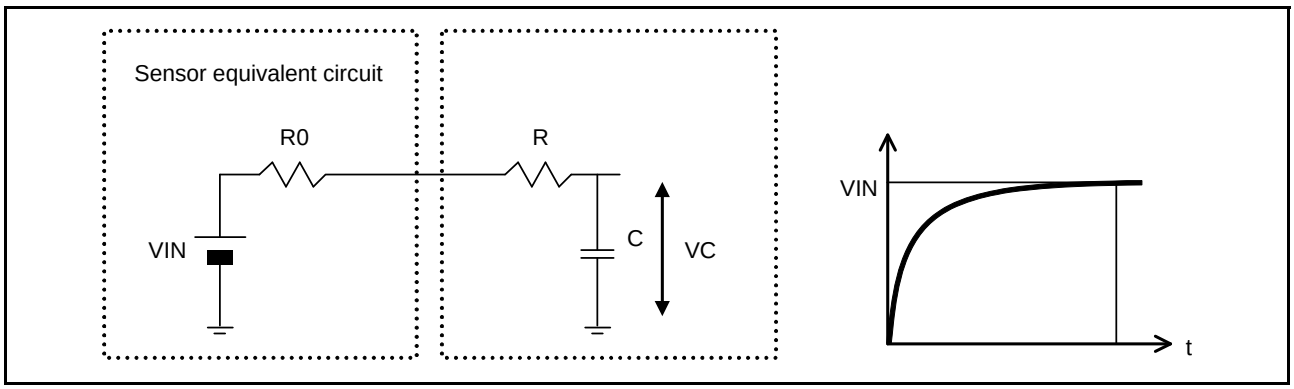



**Figure 30.15 Equivalent Circuit of Analog Input Pin and External Sensor**

Let the sampling time be T [s], the output impedance of the sensor be R0 [Ω], the internal resistance of the MCU be R [Ω], the accuracy (error) of the A/D converter be x [LSB], and the resolution of the A/D converter be y [tone] (4096 in 12-bit mode), the voltage difference VC between both sides of the capacitor C is expressed as follows:

$$VC = VIN\left\{1 - e^{-\frac{T}{C(R0+R)}}\right\}$$

When t = T, to obtain a conversion error of x or less,

$$VC = VIN - \frac{x}{y}VIN = VIN\left(1 - \frac{x}{y}\right)$$

Thus, R0 can be calculated as follows:

$$e^{-\frac{T}{C(R0+R)}} = \frac{x}{y}$$

$$-\frac{T}{C(R0+R)} = 1n\frac{x}{y}$$

$$R0 = -\frac{T}{C1n\frac{x}{y}} - R$$

When fPCLKD = 32 MHz, the output impedance R0 of the sensor for an error of 0.1 LSB or less is obtained by the following equation:
T = 0.3 µs, x = 0.1, y = 4096, R = 2.6 kΩ (reference value), and C = 7 pF (reference value)
Hence,

$$R0 = -\frac{0.3\times10^{-6}}{7\times10^{-12}\times1n\left(\frac{0.1}{4096}\right)} - 2.6\times10^{3}$$

$$= 1435$$

Thus, the output impedance R0 of the sensor must be approximately 1.4 kΩ or less to obtain the A/D converter accuracy (error) of 0.1 LSB or less.
Actual error, however, is the value of absolute accuracy added to the above 0.1 LSB.
These values are for reference and operation must be verified by performing evaluations.

### 30.7.9 Influence on Absolute Accuracy

Attaching a capacitor creates coupling with GND and may affect the absolute accuracy when noisy GND is used; therefore, a capacitor should be connected to electrically stable GND such as AVSS0.
The filter circuit should be designed so that it does not interfere digital signals or it does not serve as an antenna on the circuit board.

### 30.7.10 Voltage Range of Analog Power Supply Pins

If this MCU is used with the voltages outside the following ranges, the reliability of the MCU may be affected.

- Analog input voltage range
  Voltage (VAN) applied to analog input pins ANn: VREFL0 ≤ VAN ≤ VREFH0
  Reference voltage range applied to the VREFH0 pin: VREFH0 ≤ AVCC0
  Voltage applied to analog input pins ANn (n = 0 to 4, 6): AVSS0 ≤ VAN ≤ AVCC0
  Voltage applied to analog input pins ANn (n = 8 to 15): VSS ≤ VAN ≤ VCC and VSS ≤ VAN ≤ AVCC
- Relationship between power supply pin pairs (AVCC0–AVSS0, VREFH0–VREFL0, VCC–VSS)
  Relationship between AVSS0 and VSS: AVSS0 = VSS. A 0.1-µF capacitor should be connected between each pair of power supply pins to create a closed loop with the shortest rout possible as shown in Figure 30.16, and connection should be made so that the following conditions are satisfied at the supply side.
  VREFL0 = AVSS0 = VSS
  When the A/D converter is not used, the following conditions should be satisfied.
  VREFH0 = AVCC0 = VCC and VREFL0 = AVSS0 = VSS

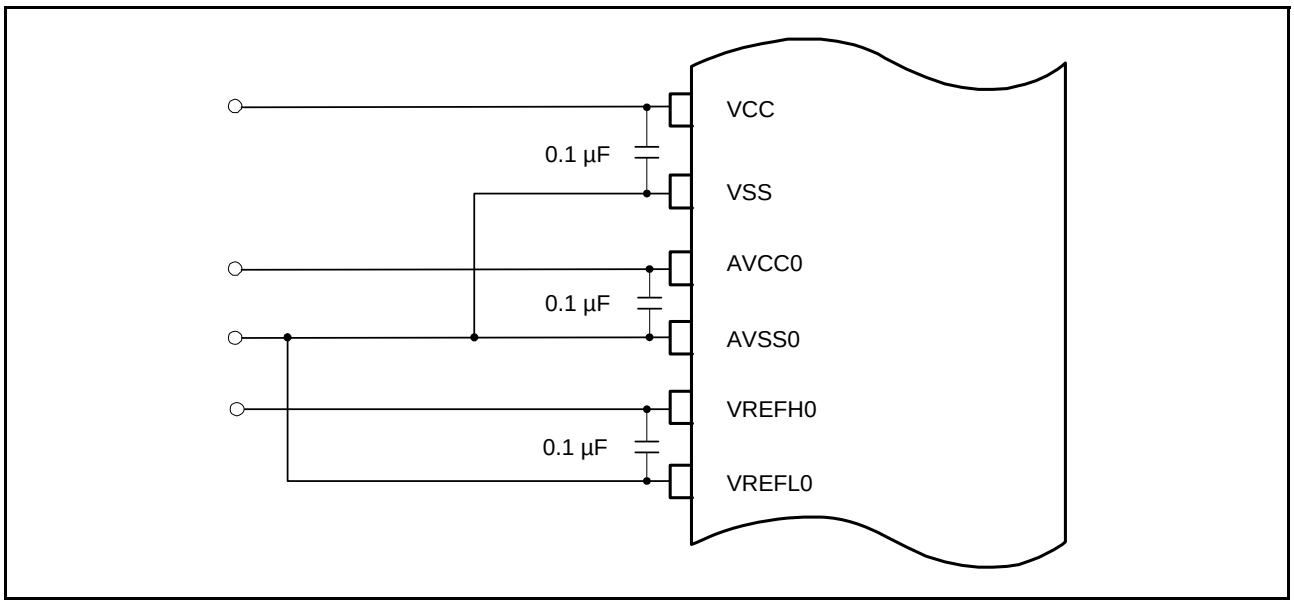


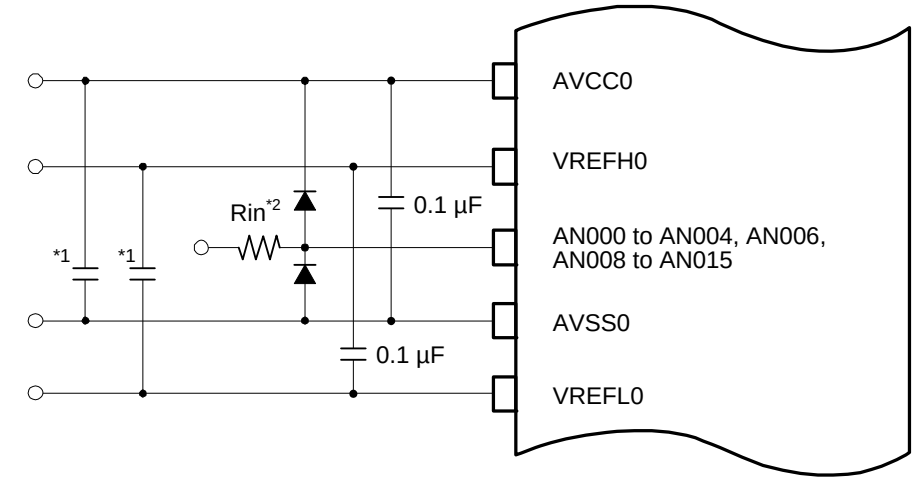

**Figure 30.16 Power Supply Pin Connection Example**

### 30.7.11 Notes on Board Design

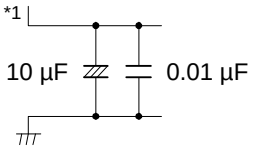
The board should be designed so that digital circuits and analog circuits are separated from each other as far as possible. In addition, digital circuit signal lines and analog circuit signal lines should not intersect or placed near each other. If these rules are not followed, noise will be produced on analog signals and A/D conversion accuracy will be affected. The analog input pins (AN000 to AN004, AN006, AN008 to AN015), reference power supply pin (VREFH0), reference ground pin (VREFL0), and analog power supply (AVCC0) should be separated from digital circuits using the analog ground (AVSS0). The analog ground (AVSS0) should be connected to a stable digital ground (VSS) on the board (single-point ground plane connection).

## 30.7.12 Notes on Noise Prevention

To prevent the analog input pins (AN000 to AN004, AN006, AN008 to AN015) from being destroyed by abnormal voltage such as excessive surge, a capacitor should be inserted between AVCC0 and AVSS0 and between VREFH0 and VREFL0, and a protection circuit should be connected to protect the analog input pins (AN000 to AN004, AN006, AN008 to AN015) as shown Figure 30.17.




Note 1. The values shown here are reference values.




Note 2. Rin: Signal source impedance

**Figure 30.17 Sample Protection Circuit for Analog Inputs**

## 30.7.13 Port Setting when 12-bit A/D Converter Inputs are Used

If any one of port 4 or port E pins is used as an analog input pin for the 12-bit A/D converter, none of port 4 output pin should be used. This is because an analog power supply is used for parts of the port 4 circuits. However, the output from the D/A converter can be used.

# 31. D/A Converter (DA)

## 31.1 Overview

This MCU includes two-channels of 8-bit D/A converter.
Table 31.1 lists the specifications of the D/A converter and Figure 31.1 shows a block diagram of the D/A converter.

**Table 31.1 D/A Converter Specifications**

| Item | Specifications |
|---|---|
| Resolution | 8 bits |
| Number of output channels | Two channels |
| Low power consumption function | Module stop state can be set for each unit. |
| Event link function (input) | DA0 conversion can be started when an event signal is input. |

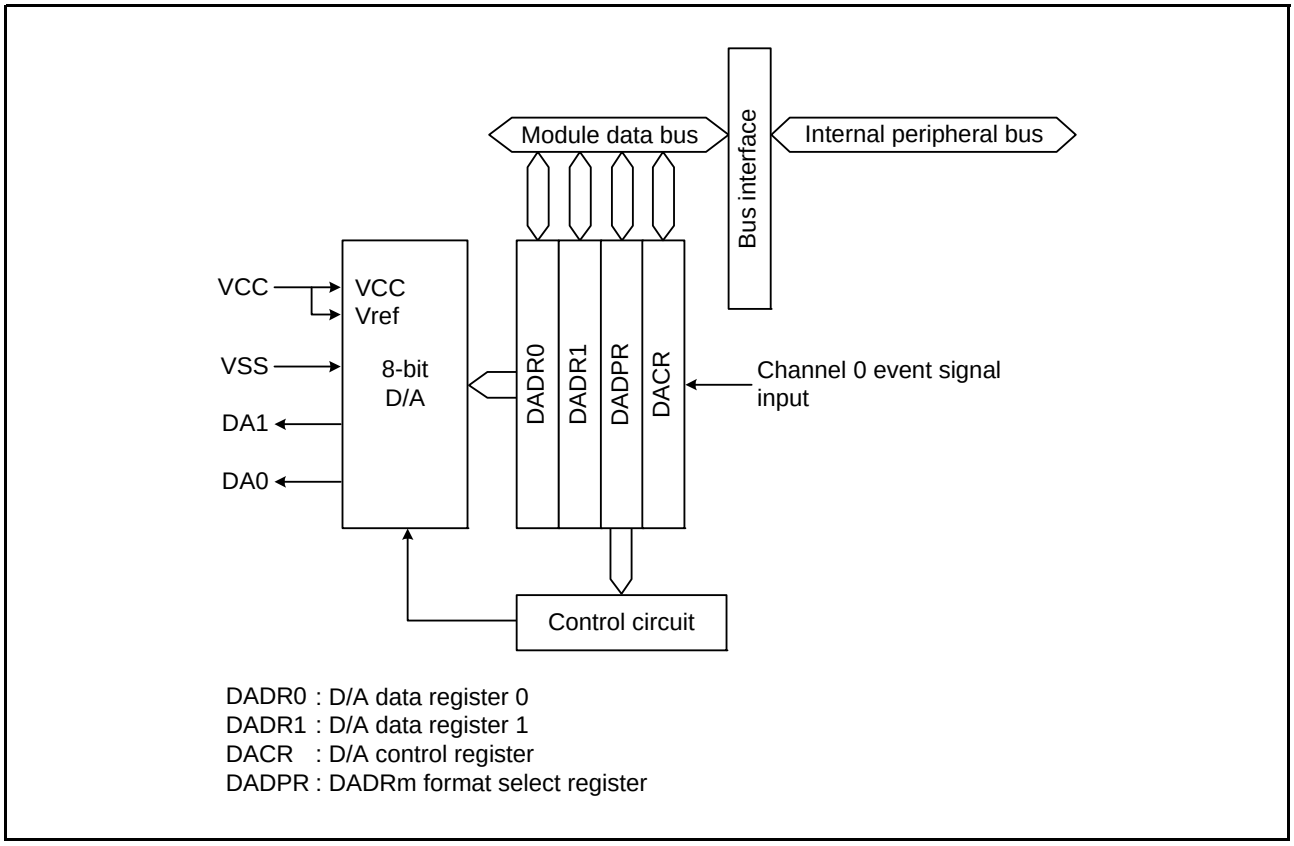



**Figure 31.1 Block Diagram of D/A Converter**

Table 31.2 lists the pin configuration of the D/A converter.

**Table 31.2 Pin Configuration of the D/A Converter**

| Pin Name | I/O | Function |
|---|---|---|
| VCC | Input | Power supply pin |
| VSS | Input | Ground pin |
| DA0 | Output | Channel 0 analog output pin |
| DA1 | Output | Channel 1 analog output pin |

## 31.2 Register Descriptions

### 31.2.1 D/A Data Register m (DADRm) (m = 0, 1)

Address(es): DADR0: 0008 80C0h, DADR1: 0008 80C2h

• DADPR.DPSEL bit = 0 (data are flush with the right end of the register)

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | | | | | | | | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

• DADPR.DPSEL bit = 1 (data are flush with the left end of the register)

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | | | — | — | — | — | — | — | — | — |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

DADRm registers are 16-bit readable/writable registers, which store data to which D/A conversion is to be performed. Whenever an analog output is enabled, the values in DADRm are converted and output to the analog output pins.
8-bit data can be relocated by setting the DADPR.DPSEL bit.
Bits “—” are read as 0. The write value should be 0.

## 31.2.2 D/A Control Register (DACR)

Address(es): 0008 80C4h

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | DAOE1 | DAOE0 | — | — | — | — | — | — |
| Value after reset: | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b4 to b0 | — | Reserved | These bits are read as 1. The write value should be 1. | R/W |
| b5 | — | Reserved | This bit is read as 0. The write value should be 0. | R/W |
| b6 | DAOE0 | D/A Output Enable 0 | 0: Analog output of channel 0 (DA0) is disabled.<br>1: D/A conversion of channel 0 is enabled.<br>Analog output of channel 0 (DA0) is enabled.*1 | R/W |
| b7 | DAOE1 | D/A Output Enable 1 | 0: Analog output of channel 1 (DA1) is disabled.<br>1: D/A conversion of channel 1 is enabled.<br>Analog output of channel 1 (DA1) is enabled.*1 | R/W |

Note 1. Set the P0.PDR.Bm bit (m = 3, 5) for pins used as analog outputs and the corresponding P0.PMR.Bm bit (m = 3, 5) to 0. In addition, set them to analog pins by the P03PFS and P05PFS registers. For details, refer to section 18, I/O Ports and section 19, Multi-Function Pin Controller (MPC).

**Table 31.3 Controls of D/A Conversion**

| b7 | b6 | |
|---|---|---|
| **DAOE1** | **DAOE0** | **Description** |
| 0 | 0 | D/A conversion of channels 0 and 1 is enabled.<br>Analog output of channels 0 and 1 (DA0, DA1) is disabled.*1 |
| | 1 | D/A conversion of channels 0 and 1 is enabled.<br>Analog output of channel 0 (DA0) is enabled. Analog output of channel 1 (DA1) is disabled.*1 |
| 1 | 0 | D/A conversion of channels 0 and 1 is enabled.<br>Analog output of channel 0 (DA0) is disabled.*1 Analog output of channel 1 (DA1) is enabled. |
| | 1 | D/A conversion of channels 0 and 1 is enabled.<br>Analog output of channels 0 and 1 (DA0, DA1) is enabled. |

Note 1. When analog output is disabled, the analog output signal is placed in the Hi-Z state.

### DAOE0 Bit (D/A Output Enable 0)

The DAOE0 bit controls D/A conversion and analog output.

The event link function is capable of setting the DAOE0 bit to 1. The DAOE0 bit becomes 1 when the event specified by the setting of the ELSR16 register of the ELC occurs, and output of the D/A conversion results starts.

### DAOE1 Bit (D/A Output Enable 1)

The DAOE1 bit controls D/A conversion and analog output.

## 31.2.3 DADRm Format Select Register (DADPR)

Address(es): 0008 80C5h

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | DPSEL | — | — | — | — | — | — | — |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b6 to b0 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |
| b7 | DPSEL | DADRm Format Select | 0: Data is flush with the right end of the D/A data register.<br>1: Data is flush with the left end of the D/A data register. | R/W |

## 31.3 Operation

The D/A converter includes D/A conversion circuits for two channels, each of which can operate independently. When the DACR.DAOEi bit (i = 0, 1) is set to 1, D/A converter is enabled and the conversion result is output.

An operation example of D/A conversion on channel 0 is shown below. Figure 31.2 shows the timing of this operation.

1. Set the DADPR.DPSEL bit and set the data for D/A conversion to the DADR0 register.
2. Set the DACR.DAOE0 bit to 1 to start D/A conversion. The conversion result is output from the DA0 pin after tDCONV elapses. The conversion result continues to be output until DADR0 is written to again or the DAOE0 bit is cleared to 0. The output value is expressed by the following formula:

$$\frac{\text{Setting value of DADR0}}{256} \times \text{VCC}$$

3. If DADR0 is written to again, the conversion starts immediately. The conversion result is output after tDCONV elapses.
4. If the DAOE0 bit is cleared to 0, analog output is disabled.

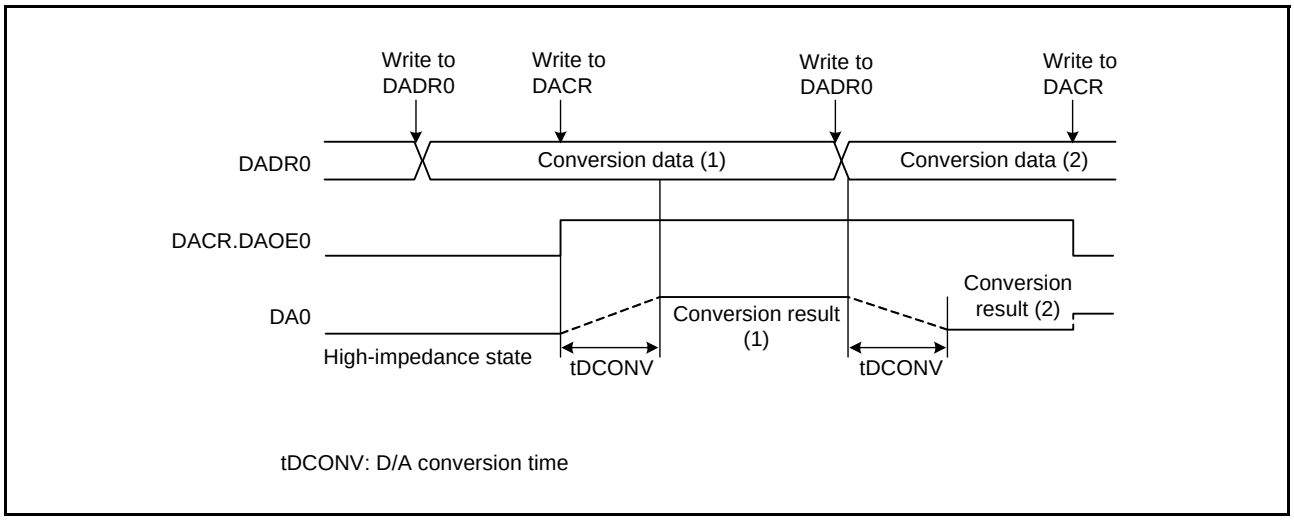



**Figure 31.2 Example of D/A Converter Operation**

## 31.4 Event Link Operation Setting Procedure

The event link operation procedure is described below.

1. Set the data to convert D/A to the DADPR.DPSEL bit setting and DADR0 register.
2. Set the bit value of the ELSR16 setting event signal to link the ELSR16 of the ELC.
3. Set the ELCR.ELCON bit to 1. This procedure enables the event link operation for all modules with the event link function selected.
4. Set the event output source module to activate the event link. After the event is output from the module, the DACR.DAOE0 bit becomes 1, and D/A conversion on channel 0 starts.
5. Set the ELSR16.ELS[7:0] bits to 0000 0000b to stop the event link operation of the D/A converter channel 0. All event link operation is stopped when the ELCR.ELCON bit is set to 0.

## 31.5 Usage Notes on Event Link Operation

When the specified event is generated by the ELSR16 register while write cycle is performed to the DACR.DAOE0 bit, the write cycle is not performed to the DACR.DAOE0 bit, and the setting to 1 takes precedence by the generated event.

## 31.6 Usage Notes

### 31.6.1 Module Stop Function Setting

Operation of the D/A converter can be disabled or enabled by setting module stop control register A (MSTPCRA). The initial setting is for operation of the D/A converter to be halted. Register access is enabled by clearing the module stop state. For details, refer to section 11, Low Power Consumption.

### 31.6.2 Operation of the D/A Converter in Module Stop State

When the module is stopped while D/A conversion is enabled, the D/A outputs are retained, and the analog power supply current becomes the same as the current during D/A conversion. If the analog power supply current has to be reduced in the module is stopped, disable D/A conversion by setting the DACR.DAOE1 and DAOE0 bits to 0.

### 31.6.3 Operation of the D/A Converter in Software Standby Mode

When the MCU enters software standby mode while D/A conversion is enabled, the D/A outputs are retained, and the analog power supply current becomes the same as the current during D/A conversion. If the analog power supply current needs to be reduced while in software standby mode, disable D/A conversion by setting the DAOE1 and DAOE0 bits to 0.

# 32. Temperature Sensor (TEMPSa)

## 32.1 Overview

This MCU includes a temperature sensor. The temperature sensor outputs a voltage which varies with the temperature. The user can obtain the temperature surrounding the MCU using the 12-bit A/D converter to convert the voltage output from the temperature sensor into a digital value.

Table 32.1 lists the specifications of the temperature sensor. Figure 32.1 shows an overall block diagram of the temperature sensor system.

**Table 32.1 Temperature Sensor Specifications**

| Item | Description |
|---|---|
| Temperature sensor voltage output | The temperature sensor voltage is output to the 12-bit A/D converter. |

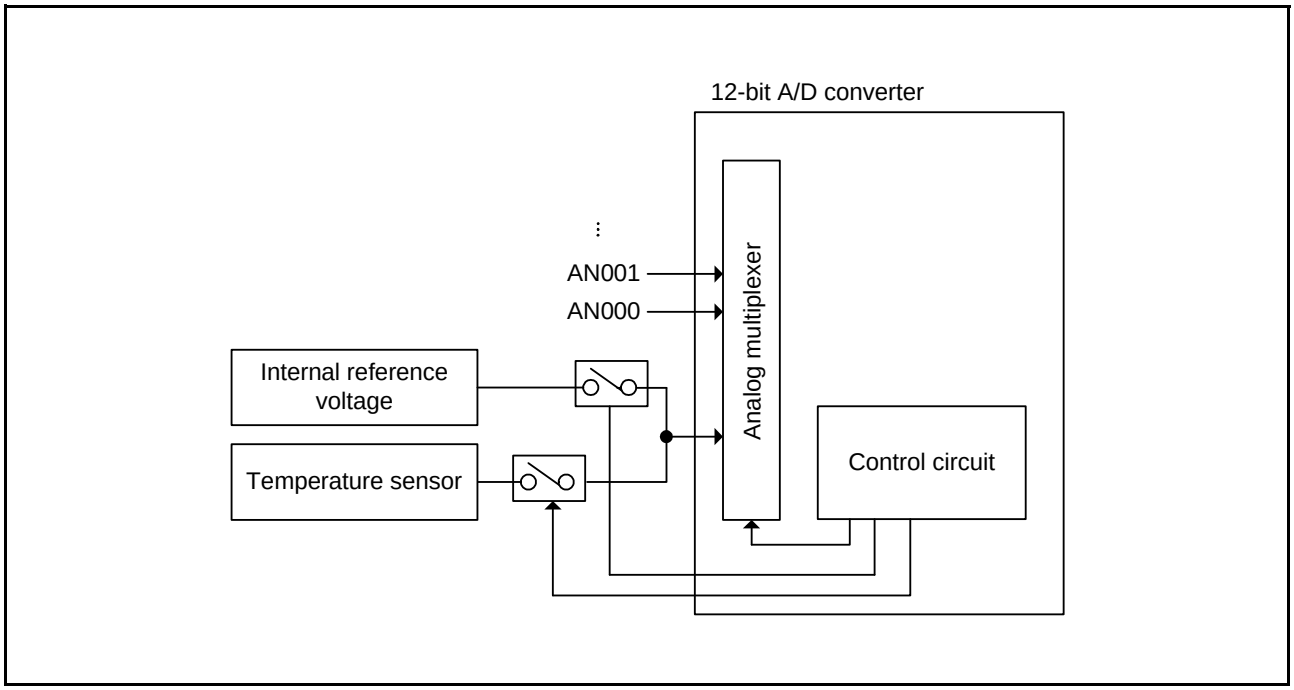



**Figure 32.1 Block Diagram of Temperature Sensor System**

## 32.2 Using the Temperature Sensor

The temperature sensor outputs a voltage which varies with the temperature. The user can obtain the temperature surrounding the MCU using the 12-bit A/D converter to convert this voltage into a digital value.

### 32.2.1 Before Using the Temperature Sensor

The temperature characteristics of the temperature sensor are shown below. The voltage output by the temperature sensor is proportional to the temperature, which can be calculated according to the formula below.

Formula for the temperature characteristic:
T = (Vs – V1)/Slope + T1

T: Measured temperature (°C)
Vs: Voltage output by the temperature sensor when the temperature is measured (V)
T1: Sample temperature measurement at first point (°C)
V1: Voltage output by the temperature sensor when T1 is measured (V)
T2: Sample temperature measurement at second point (°C)
V2: Voltage output by the temperature sensor when T2 is measured (V)
(V2 - V1)/(T2 - T1) = Slope: Temperature gradient of the temperature sensor (V/°C)

Characteristics vary from sensor to sensor. Therefore, it is recommended that two different sample temperatures are measured.
Use the 12-bit A/D converter to measure the voltage V1 output by the temperature sensor at temperature T1.
Again, using the 12-bit A/D converter, measure the voltage V2 output by the temperature sensor at a different temperature T2. Obtain the temperature gradient (Slope = (V2 – V1)/(T2 – T1)) from these results.
Subsequently, obtain temperatures by substituting the slope into the formula for the temperature characteristic (T = (Vs – V1)/Slope + T1).

If you are using the temperature gradient given in section 36., Electrical Characteristics, use the A/D converter to measure the voltage V1 output by the temperature sensor at temperature T1, and then calculate the temperature characteristic by using the formula below.
However, this method produces less accurate temperatures than measurement at two points.

T = (Vs – V1)/Slope + T1

### 32.2.2 Setting the 12-bit A/D Converter

The temperature sensor can provide temperature data through the A/D conversion of the temperature sensor output. In order to A/D convert output from the temperature sensor, 12-bit A/D converter registers must be set as follows.

- Selecting the temperature sensor voltage as an A/D conversion target
  Set the ADEXICR.TSS bit to 1 to A/D convert temperature sensor output. Additionally, remove other sources from the scope of conversion by setting the ADEXICR.OCS bit, and all bits in registers ADANSA, ADANSB, and ADEXICR.OCS to 0.
- Setting single scan mode
  Set the ADCSR.ADCS[1:0] bits to 00 to select single scan mode. Do not select the other mode.

### 32.2.3 A/D Conversion Result of Temperature Sensor Output

After the temperature sensor output is A/D converted, the conversion result is stored in the ADTSDR register.
Set the sampling time to 5 µs or longer. After switching to A/D conversion of the temperature sensor output, set the ADST bit to 1 and start the first conversion. However, do not use the first conversion result. Figure 32.2 shows an example of operating the temperature sensor.

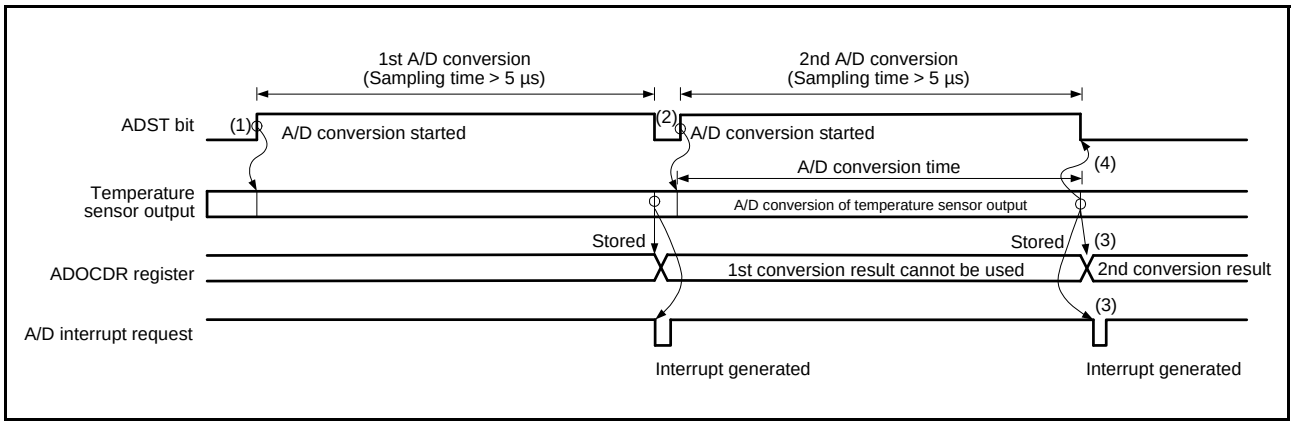



**Figure 32.2 Example of Temperature Sensor Operation**

# 33. Data Operation Circuit (DOC)

## 33.1 Overview

The data operation circuit (DOC) is used to compare, add, and subtract 16-bit data.
Table 33.1 lists the data operation circuit specifications and Figure 33.1 shows a block diagram of the data operation circuit.
16-bit data is compared and an interrupt can be generated when a selected condition applies.

**Table 33.1 Specifications of Data Operation Circuit**

| Item | Description |
|---|---|
| Data operation function | 16-bit data comparison, addition, and subtraction |
| Power consumption reduction function | Module stop state can be set. |
| Event link function (output) | • An interrupt occurs when the compared values either match or mismatch.<br>• The result of data addition being greater than FFFFh<br>• The result of data subtraction being less than 0000h |

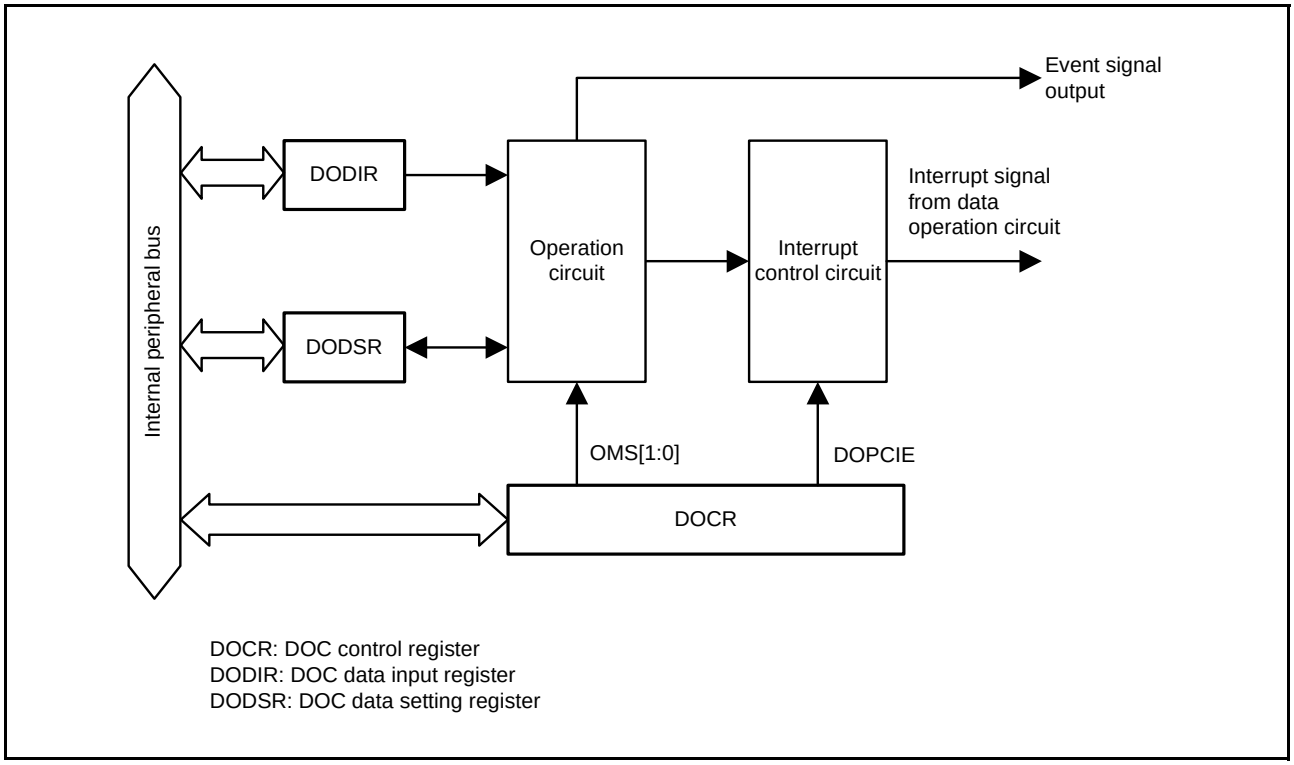



**Figure 33.1 Block Diagram of Data Operation Circuit**

## 33.2 Register Descriptions

### 33.2.1 DOC Control Register (DOCR)

Address(es): 0008 B080h

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | DOPCFCL | DOPCF | DOPCIE | — | DCSEL | OMS[1:0] | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b1, b0 | OMS[1:0] | Operating Mode Select | b1 b0<br>0 0: Data comparison mode<br>0 1: Data addition mode<br>1 0: Data subtraction mode<br>1 1: Setting prohibited | R/W |
| b2 | DCSEL*1 | Detection Condition Select | Result of data comparison<br>0: Data mismatch is detected.<br>1: Data match is detected. | R/W |
| b3 | — | Reserved | This bit is read as 0. The write value should be 0. | R/W |
| b4 | DOPCIE | Data Operation Circuit Interrupt Enable | 0: Disables interrupts from the data operation circuit.<br>1: Enables interrupts from the data operation circuit. | R/W |
| b5 | DOPCF | Data Operation Circuit Flag | Indicates the result of an operation. | R |
| b6 | DOPCFCL | DOPCF Clear | 0: Maintains the DOPCF flag state.<br>1: Clears the DOPCF flag. | R/W |
| b7 | — | Reserved | This bit is read as 0. The write value should be 0. | R/W |

Note 1. Valid only when data comparison mode is selected.

#### OMS[1:0] Bits (Operating Mode Select)

These bits select the operating mode of the data operation circuit.

#### DCSEL Bit (Detection Condition Select)

This bit is valid only when data comparison mode is selected.

This bit selects the condition for detection in data comparison mode.

#### DOPCIE Bit (Data Operation Circuit Interrupt Enable)

Setting this bit to 1 enables interrupts from the data operation circuit.

#### DOPCF Flag (Data Operation Circuit Flag)

[Setting conditions]

- The condition selected by the DCSEL bit being met
- A result of data addition being greater than FFFFh
- A result of data subtraction being less than 0000h

[Clearing condition]

- Writing 1 to the DOPCFCL bit

#### DOPCFCL Bit (DOPCF Clear)

Setting this bit to 1 clears the DOPCF flag.

This bit is read as 0.

### 33.2.2 DOC Data Input Register (DODIR)

Address(es): 0008 B082h

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

DODIR is a 16-bit readable/writable register in which 16-bit data for use in the operations are stored.

### 33.2.3 DOC Data Setting Register (DODSR)

Address(es): 0008 B084h

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

DODSR is a 16-bit readable/writable register. This register stores 16-bit data for use as a reference in data comparison mode. This register also stores the results of operations in data addition and data subtraction modes.

## 33.3 Operation

### 33.3.1 Data Comparison Mode

Figure 33.2 shows an example of the steps involved in data comparison mode operation by the data operation circuit. The following is an example of operation when DCSEL is set to 0 (detects mismatch as a result of data comparison).

(1) Writing 00b to the DOCR.OMS[1:0] bits selects data comparison mode.
(2) The 16-bit reference data is set in DODSR.
(3) 16-bit data for comparison is written to DODIR.
(4) Writing of 16-bit data continues until all data for comparison have been written to DODIR.
(5) If a value written to DODIR does not match that in DODSR*1, the DOCR.DOPCF flag is set to 1. When the DOCR.DOPCIE bit is 1, a data operation circuit interrupt is also generated.

Note 1. When DOCR.DCSEL = 0

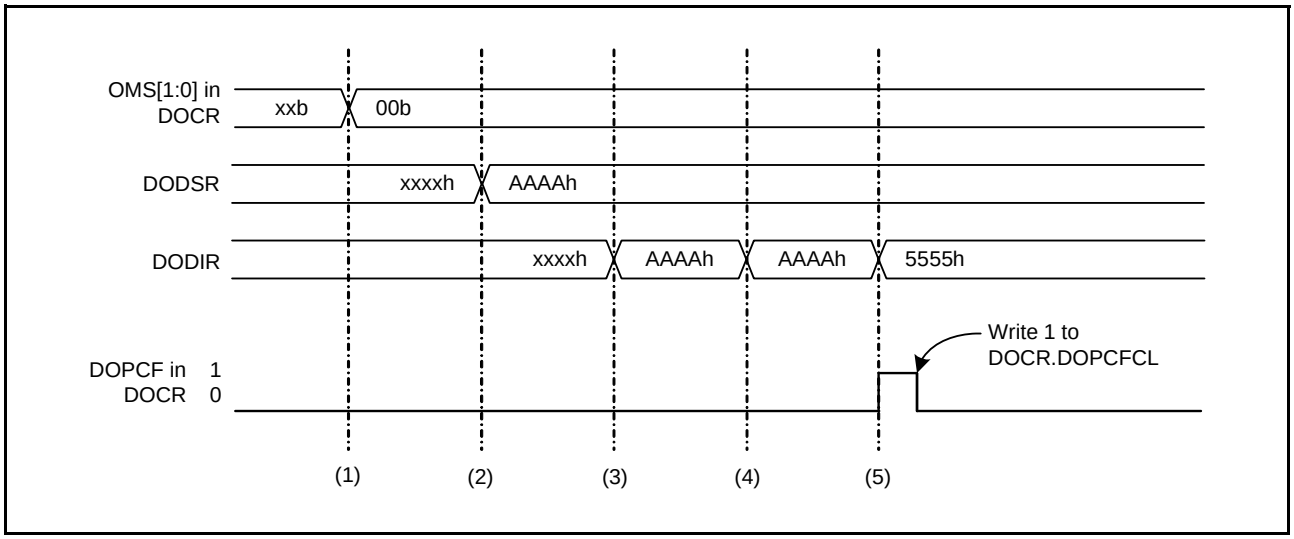



**Figure 33.2 Example of Operation in Data Comparison Mode**

### 33.3.2 Data Addition Mode

Figure 33.3 shows an example of the steps involved in data addition mode operation by the data operation circuit.

(1) Writing 01b to the DOCR.OMS[1:0] bits selects data addition mode.
(2) 16-bit data is set in the DODSR register as the initial value.
(3) 16-bit data to be added is written to DODIR. The result of the operation is stored in DODSR.
(4) Writing of 16-bit data continues until all data for addition have been written to DODIR.
(5) If the result of an operation is greater than FFFFh, the DOCR.DOPCF flag is set to 1. When the DOCR.DOPCIE bit is 1, a data operation circuit interrupt is also generated.

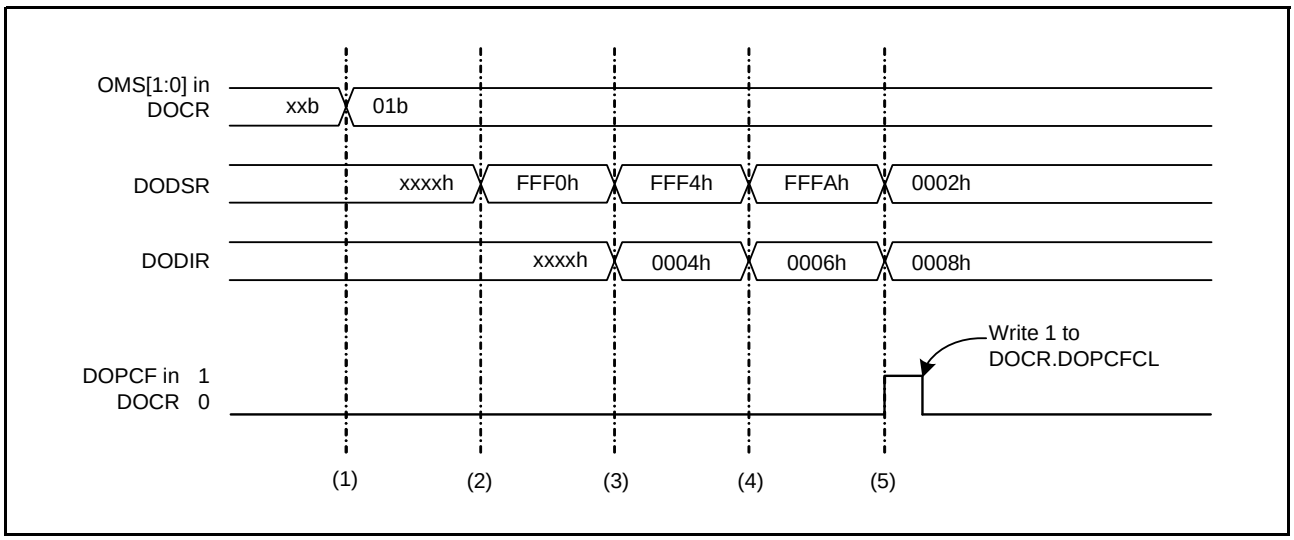



**Figure 33.3 Example of Operation in Data Addition Mode**

### 33.3.3 Data Subtraction Mode

Figure 33.4 shows an example of the steps involved in data subtraction mode operation by the data operation circuit.

(1) Writing 10b to the DOCR.OMS[1:0] bits selects data subtraction mode.
(2) 16-bit data is set in the DODSR register as the initial value.
(3) 16-bit data to be subtracted is written to DODIR. The result of the operation is stored in DODSR.
(4) Writing of 16-bit data continues until all data for subtraction have been written to DODIR.
(5) If the result of an operation is less than 0000h, the DOCR.DOPCF flag is set to 1. When the DOCR.DOPCIE bit is 1, a data operation circuit interrupt is also generated.

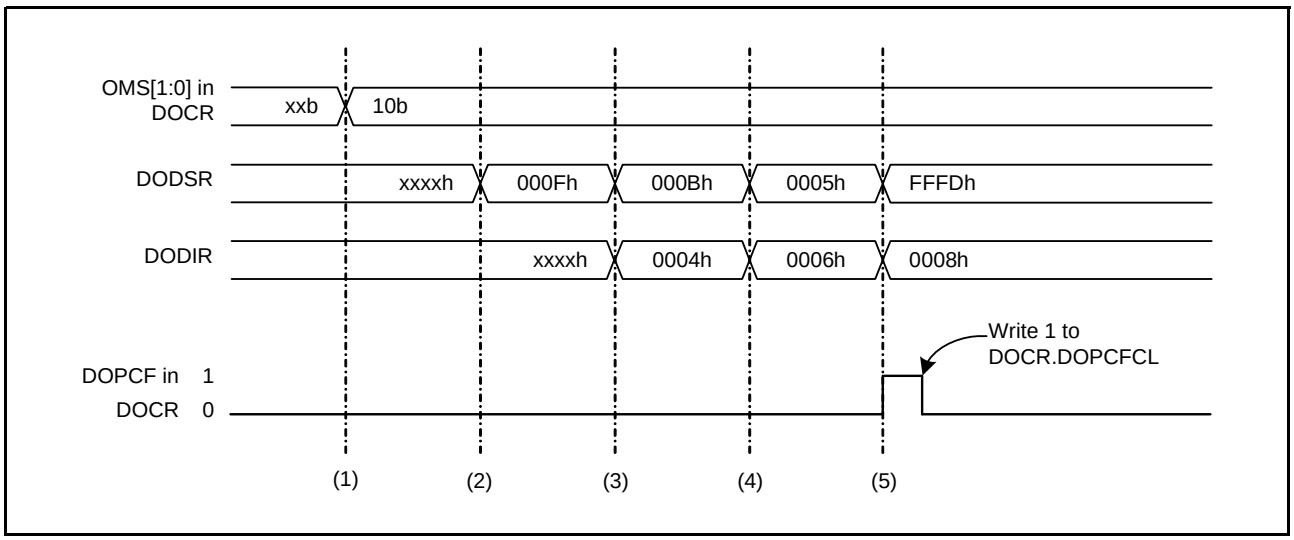



**Figure 33.4 Example of Operation in Data Subtraction Mode**

## 33.4 Interrupt Requests

The data operation circuit generates the data operation circuit interrupt as an interrupt request. When an interrupt source arises, the data operation circuit flag corresponding to the interrupt is set to 1. Table 33.2 describes the interrupt request.

**Table 33.2 Interrupt Request from Data Operation Circuit**

| Interrupt Request | Data Operation Circuit Flag | Interrupt Generation Timing |
|---|---|---|
| Data operation circuit interrupt | DOPCF | • An interrupt occurs when the compared values either match or mismatch.<br>• The result of data addition being greater than FFFFh<br>• The result of data subtraction being less than 0000h |

## 33.5 Event Link Output

The DOC outputs event signals for the event link controller (ELC) under the following conditions, and these can be used to initiate operations by other modules selected in advance.

- An interrupt occurs when the compared values either match or mismatch.
- The result of data addition being greater than FFFFh
- The result of data subtraction being less than 0000h

### 33.5.1 Interrupt Handling and Event Linking

The DOC has a bit to enable or disable interrupts. An interrupt request signal is output for the CPU when an interrupt source is generated while the corresponding enable bit is enabled.
In contrast, an event link output signal is sent to other modules as an event signal via the ELC when an interrupt source is generated, regardless of the setting of the corresponding interrupt enable bit.

## 33.6 Usage Note

### 33.6.1 Module Stop Function Setting

Operation of the data operation circuit can be disabled or enabled using module stop control register B (MSTPCRB). The initial setting is for the data operation circuit to be stopped. Register access is enabled by canceling the module stop state. For details, see section 11, Low Power Consumption.

# 34. RAM

This MCU has an on-chip high-speed static RAM.

## 34.1 Overview

Table 34.1 lists the specifications of the RAM.

**Table 34.1 RAM Specifications**

| Item | Description |
|---|---|
| RAM capacity | Max. 16 Kbytes*2 |
| Access | • Single-cycle access is possible for both reading and writing.<br>• RAM can be enabled or disabled.*1 |
| Low power consumption function | Module stop state can be set. |

Note 1. Selectable by the SYSCR1.RAME bit. For details on the SYSCR1 register, refer to section 3.2.2, System Control Register 1 (SYSCR1).

Note 2. The capacity of RAM differs depending on the products.

| RAM Capacity | RAM Address |
|---|---|
| 16 Kbytes | RAM0: 0000 0000h to 0000 3FFFh |
| 10 Kbytes | RAM0: 0000 0000h to 0000 27FFh |
| 8 Kbytes | RAM0: 0000 0000h to 0000 1FFFh |

## 34.2 Operation

### 34.2.1 Low Power Consumption Function

Power consumption can be reduced by setting module stop control register C (MSTPCRC) to stop supply of the clock signal to the RAM.

Setting the MSTPCRC.MSTPC0 bit to 1 stops supply of the clock signal to RAM0.

Stopping supply of the clock signal places the RAM0 in the module stop state. The RAM operates after the value is initialized by a reset.

The RAM is not accessible in the module stop state. Do not make a transition to the module stop state while the RAM is being accessed.

For details on the MSTPCRC register, refer to section 11, Low Power Consumption.

# 35. Flash Memory

This MCU has packages with a 16, 32, 64, 96, and 128 Kbyte flash memory (ROM) for storing code and 8-Kbyte flash memory (E2 DataFlash) for storing data.

## 35.1 Overview

Table 35.1 lists the specifications of the flash memory.

**Table 35.1 Flash Memory Specifications**

| Item | Description |
|---|---|
| Memory space | • User area: Up to 128 Kbytes<br>• Data area: 8 Kbytes |
| Software commands | • The following commands can be executed in boot mode or during self-programming: blank check, block erase, program, read, set access window<br>• Checksum can be also executed in boot mode. Suspend/resume can be also executed during self-programming. |
| On-board programming | SCI mode in boot mode *1<br>• Channel 1 of the serial communications interface (SCI1) is used for asynchronous serial communication.<br>• The user area and data area are rewritable.<br>USB interface mode in boot mode *1<br>• Channel 0 of the USB 2.0 function (USB0) module is used.<br>• The user area and data area are rewritable.<br>• The flash memory can be rewritable in self-powered or bus-powered mode.<br>• A personal computer can be connected using only a USB cable.<br>Self-programming in single-chip mode<br>• The user area and data area are rewritable using the self-programming library. *2 |
| Off-board programming | The user area and data area are rewritable using a flash programmer compatible with this MCU. |
| ID code protect | Connection with the flash programmer can be enabled or disabled using ID codes in boot mode. |
| Start-up program protection | This function is used to safely rewrite block 0 to block 15. |
| Area protection | This function enables rewriting only the selected blocks in the user area and disables the other blocks during self-programming. |
| Background Operation (BGO) | Programs on the ROM can be executed while rewriting the E2 DataFlash. |

Note 1. Refer to “PG-FP5 Flash Memory Programmer User’s Manual” and “Renesas Flash Programmer Flash memory programming software User’s Manual” for more details.

Note 2. The library used for self-programming is provided. Refer to section 35.10, Rewriting by Self-Programming for details on the self-programming library.

## 35.2 ROM Area and Block Configuration

A maximum of 128 Kbytes can be configured in the ROM area. The ROM area is divided into blocks according to the ROM capacity. The ROM area is erased in block units. Figure 35.1 shows the ROM Area and Block Configuration.

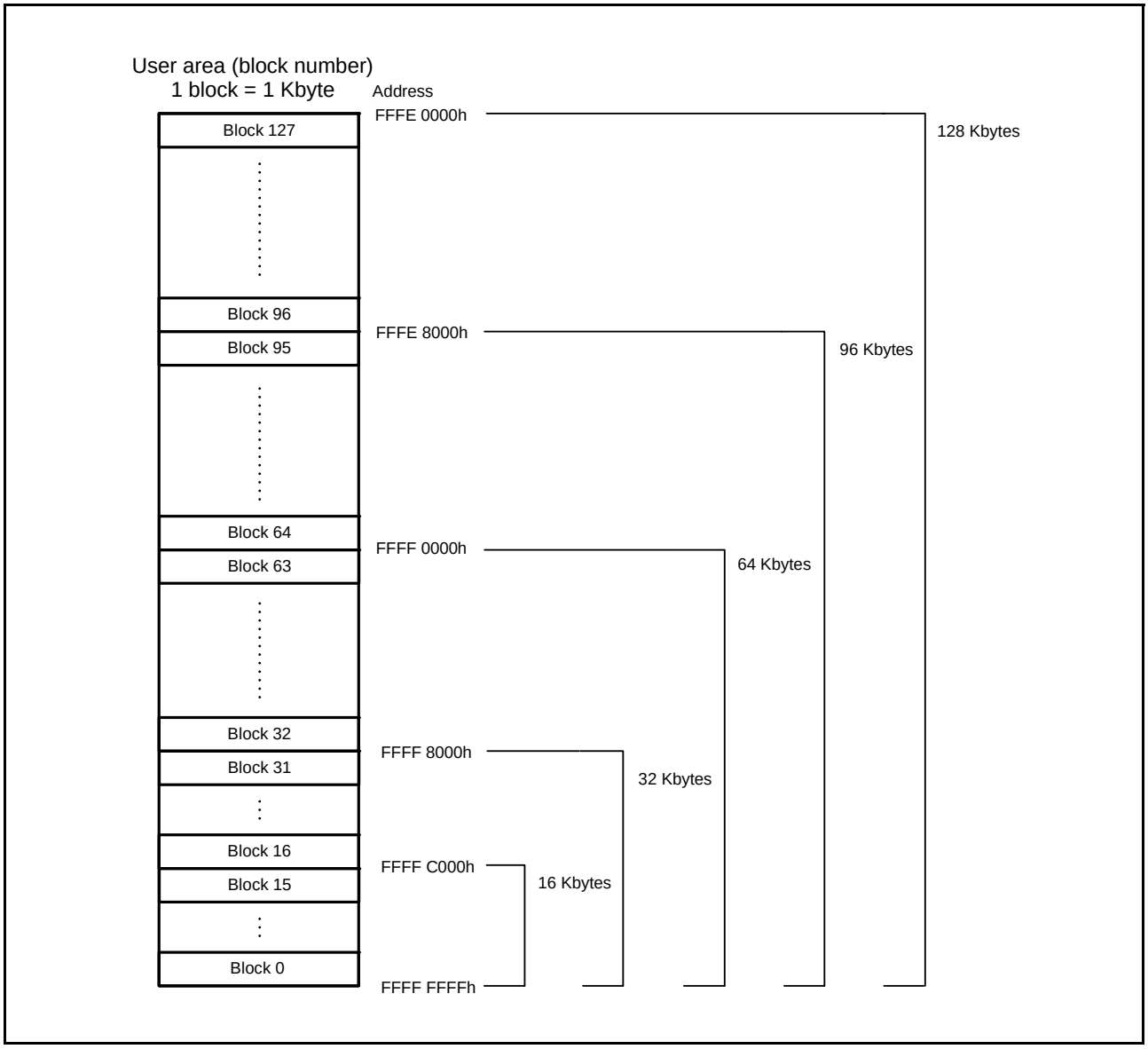



**Figure 35.1 ROM Area and Block Configuration**

**Table 35.2 Correspondence Between User Area Capacity and Addresses**

| User Area Capacity | Address |
|---|---|
| 128 Kbytes | FFFE 0000h to FFFF FFFFh |
| 96 Kbytes | FFFE 8000h to FFFF FFFFh |
| 64 Kbytes | FFFF 0000h to FFFF FFFFh |
| 32 Kbytes | FFFF 8000h to FFFF FFFFh |
| 16 Kbytes | FFFF C000h to FFFF FFFFh |

## 35.3 E2 DataFlash Area and Block Configuration

The E2 DataFlash is 8 Kbytes in the MCU. The E2 DataFlash is divided into blocks and erased in block units. Figure 35.2 shows the E2 DataFlash Area and Block Configuration.

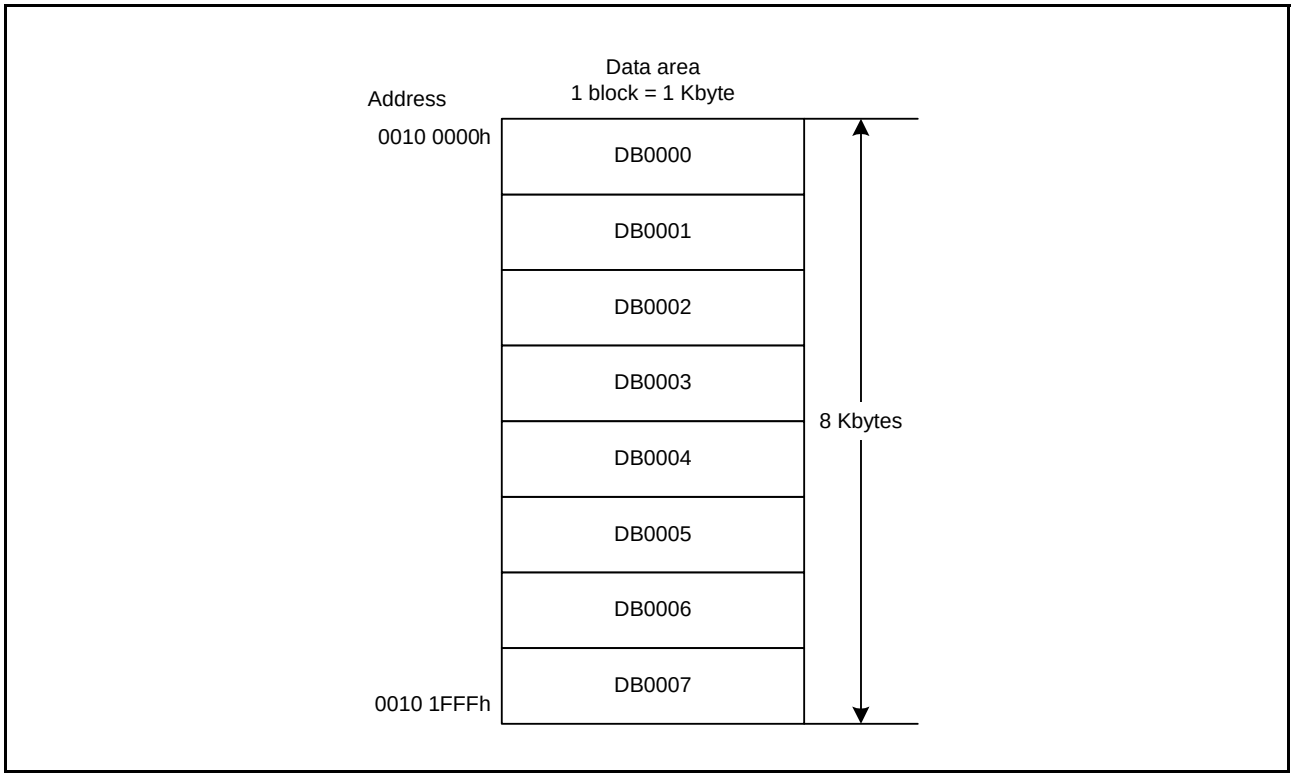



**Figure 35.2 E2 DataFlash Area and Block Configuration**

## 35.4 Register Descriptions

### 35.4.1 E2 DataFlash Control Register (DFLCTL)

Address(es): 007F C090h

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | DFLEN |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| b0 | DFLEN | E2 DataFlash Access Enable | 0: The E2 DataFlash cannot be read, programmed, and erased.<br>1: The E2 DataFlash can be read, programmed, and erased. | R/W |
| b7 to b1 | — | Reserved | These bits are read as 0. The write value should be 0. | R/W |

The DFLCTL register enables or disables reading, programming, and erasing the data area.
When reading, programming, and erasing the data area, set the DFLCTL.DFLEN bit to 1 and wait for the DataFlash STOP recovery time (tDSTOP) to elapse before reading, programming, and erasing the data area. Do not read, program, or erase the data area until tDSTOP has elapsed.
Refer to section 36., Electrical Characteristics for DataFlash STOP recovery time (tDSTOP).

## 35.5 Start-Up Program Protection

When rewriting the start-up program *1 by self-programming, if the rewrite operation is interrupted due to temporary blackout, the start-up program may not be successfully programmed and the user program may not start properly. This problem can be avoided by rewriting the start-up program without erasing the existing start-up program using the start-up program protection. This function is available in products with a 32-Kbyte or larger ROM.
Figure 35.3 shows the Overview of the Start-Up Program Protection. In this figure, Area X indicates block 0 to block 15, and Area Y indicates block 16 to block 31.

Note 1. Program to perform operation to start the user program. It includes the fixed vector table.

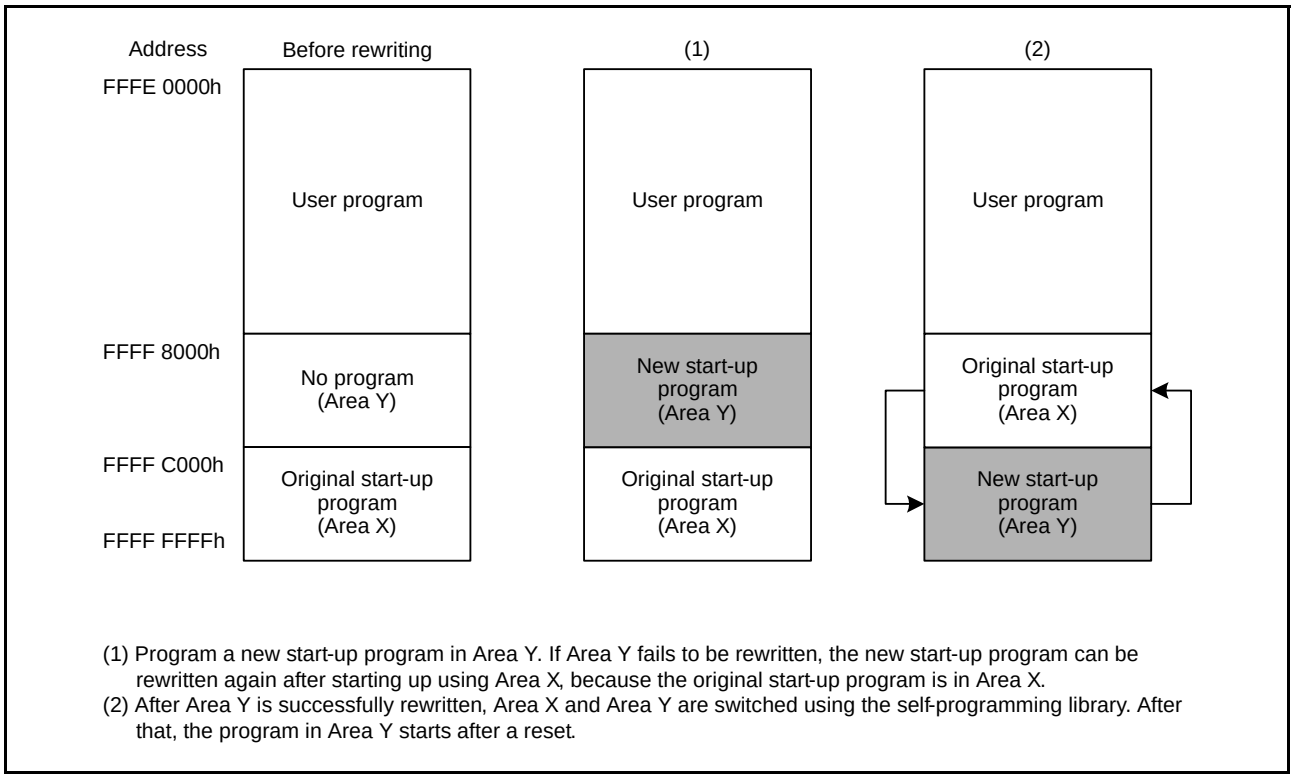



**Figure 35.3 Overview of the Start-Up Program Protection**

## 35.6 Area Protection

Area protection enables rewriting only the selected blocks (access window) in the user area and disables rewriting the other blocks during self-programming. The access window cannot be set in the data area.
Select the start block and end block to set the access window. While the access window can be set in boot mode or by self-programming, area protection is enabled only during self-programming in single-chip mode.
Figure 35.4 shows the Area Protection Overview.

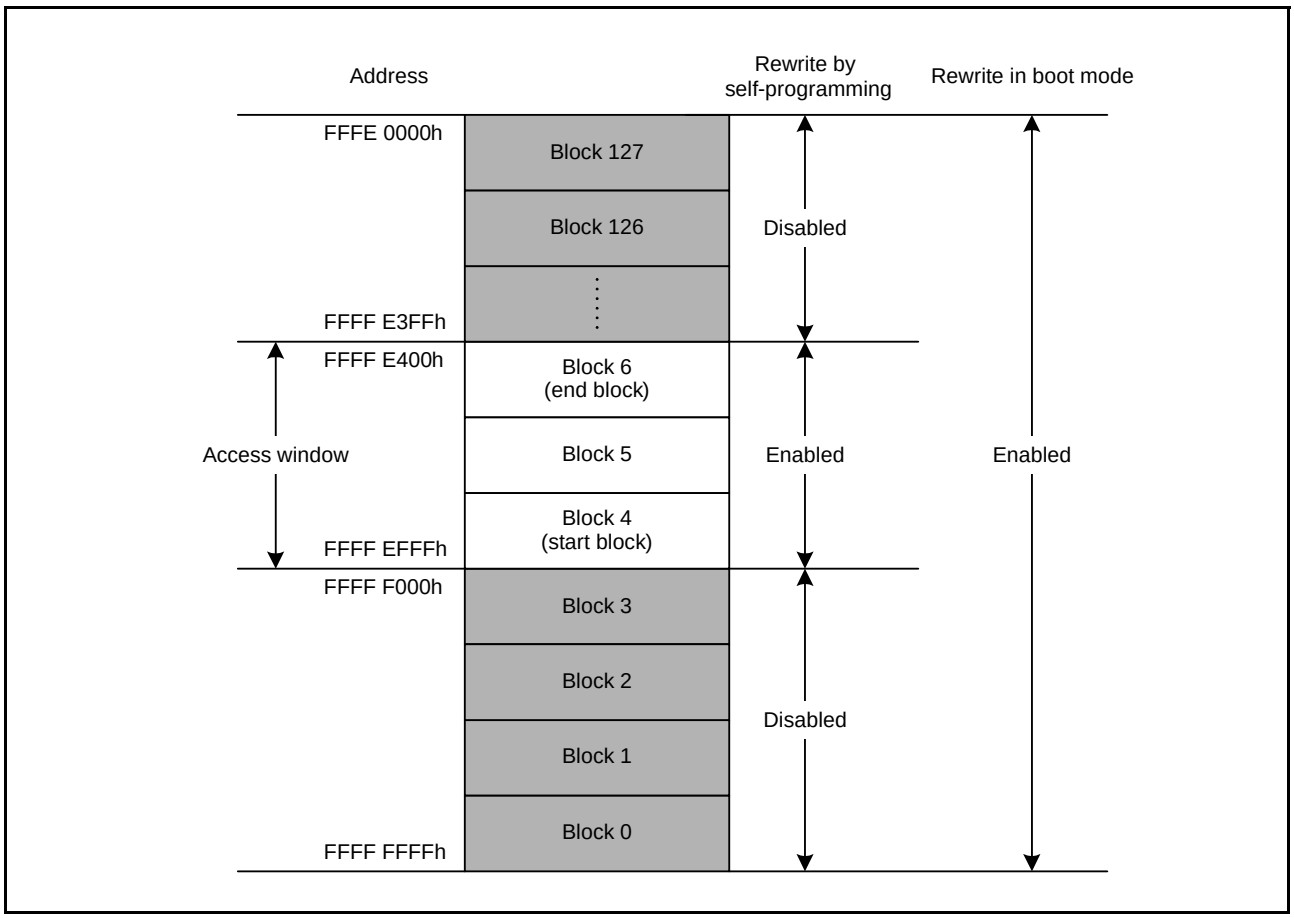



**Figure 35.4 Area Protection Overview**

## 35.7 Boot Modes

Boot mode includes USB interface mode and SCI mode.
Table 35.3 lists the Differences Between Modes. Table 35.4 lists the I/O Pins Used in Boot Mode.

**Table 35.3 Differences Between Modes**

| Item | Boot mode | |
|---|---|---|
| | **USB Interface Mode** | **SCI Mode** |
| Programmable and erasable areas | User area<br>Data area | User area<br>Data area |
| Communication function | USB0 | SCI1 (asynchronous serial communication) |

**Table 35.4 I/O Pins Used in Boot Mode**

| Pin Name | I/O | Mode | Description |
|---|---|---|---|
| P14/UB# *1 | Input | Boot mode | Select operating mode (refer to section 3., Operating Modes). |
| MD | Input | | Select operating mode (refer to section 3., Operating Modes). |
| USB0_DP, USB0_DM | I/O | USB interface mode | USB data I/O |
| P16/USB0_VBUS | Input | | Detect USB cable connection/disconnection |
| P35 | Input | | Set bus-powered mode or self-powered mode |
| P15/RXD1 | Input | SCI mode | Receive data through SCI1 for programmer communication *2 |
| P16/TXD1 | Output | | Transmit data through SCI1 for programmer communication *2 |

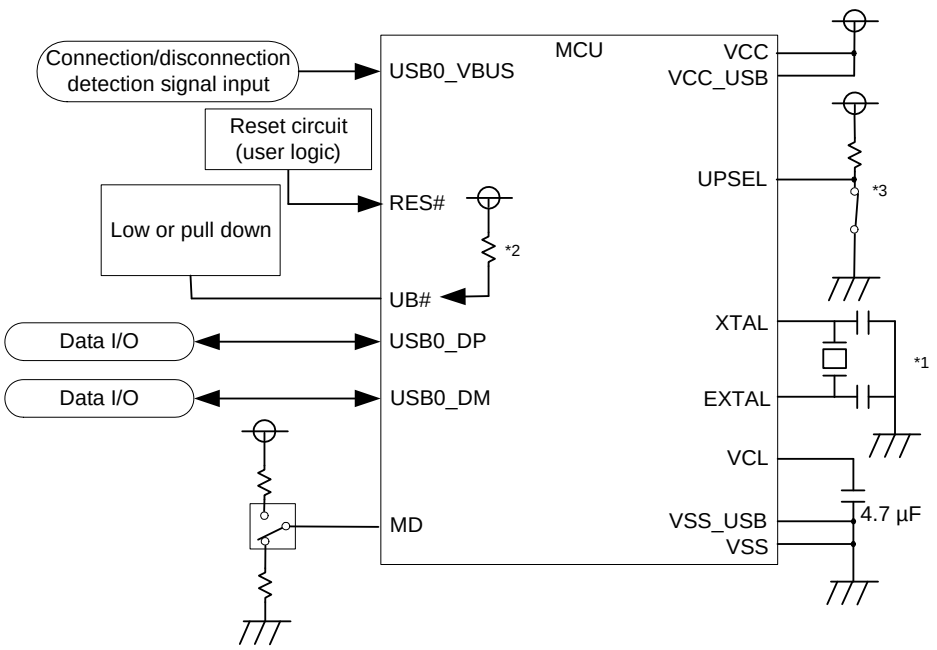
Note 1. This pin becomes high for a certain time by the on-chip input pull-up resistor after starting up in boot mode.
Note 2. When using SCI mode, connect (pull up) this pin to VCC via a resistor.

### 35.7.1 USB Interface Mode

The user area and data area can be programmed and erased using USB communication in USB interface mode.
Self power or bus power can be selected in USB interface mode in accordance of the state of the UPSEL pin.
When a reset is released while pins MD, UB#, and UPSEL are low, self power is selected. When a reset is released while pins MD and UB# are low and UPSEL is high, bus power is selected.
Contact the manufacturer for details on the dedicated flash memory programmer (USB programmer).

#### 35.7.1.1 System Configuration in USB Interface Mode

USB0 is used for communication with the dedicated flash memory programmer in USB interface mode.
Prepare tools for transmitting and receiving control commands and status via the USB and data for programming. 6 MHz, 8 MHz, 12 MHz, or 16 MHz can be used as the frequency input to the main clock oscillator. The operating voltage range is 3.0 to 3.6 V.
Connect the UB# pin to VSS on the dedicated flash memory programmer, or connect to VSS via a resistor (pull down). For pull down, connect a resistor of up to 2.5 kΩ.
Figure 35.5 shows the Example of Pin Connections in USB Interface Mode when Self-Powered. Table 35.5 lists Pin Handling in USB Interface Mode When Self-Powered. Figure 35.6 shows the Example of Pin Connections in USB Interface Mode When Bus-Powered. Table 35.6 lists Pin Handling in USB Interface Mode in Bus-Powered Mode.
Pins in Figure 35.5 show only those pins associated with USB interface mode. Operations are not guaranteed in all systems.

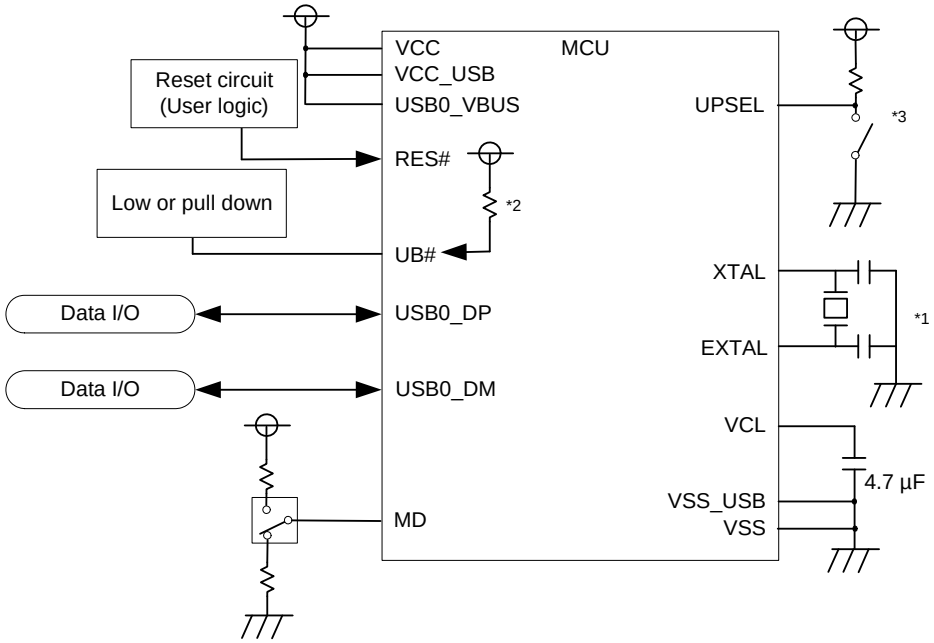



Note 1. Connect a 6 MHz, 8 MHz, 12 MHz, 16 MHz crystal, ceramic resonator or oscillator to the main clock oscillator.
Note 2. The pin is pulled up by the on-chip input pull-up resistor for a certain period after start-up in boot mode.
Note 3. UPSEL is assigned to the same pin as NMI. This is an example when the falling edge is selected by the NMI detection set bit.

**Figure 35.5 Example of Pin Connections in USB Interface Mode when Self-Powered**

**Table 35.5 Pin Handling in USB Interface Mode When Self-Powered**

| Pin Name | Name | I/O | Function |
|---|---|---|---|
| VCC, VSS | Power input | Input | Input the guaranteed voltage for program/erase to the VCC pin. Input 0 V to the VSS pin. |
| VCC_USB, VSS_USB | USB Power input | Input | Connect the VCC_USB pin to the VCC pin. Connect the VSS_USB pin to the VSS pin. |
| AVCC0, AVSS0 | 12-bit A/D converter power input | Input | Connect the AVCC0 pin to the VCC pin. Connect the AVSS0 pin to the VSS pin. |
| VREFH0, VREFL0 | 12-bit A/D converter reference power input | Input | Connect the VREFH0 pin to the VCC pin. Connect the VREFL0 pin to the VSS pin. |
| VCL | Decoupling capacitor connect pin | — | Connect to the VSS pin via a decoupling capacitor (4.7 µF) for stabilizing the internal voltage. |
| XTAL, EXTAL | Main clock I/O pin | I/O | Connect a 6 MHz, 8 MHz, 12 MHz, or 16 MHz crystal or ceramic resonator or oscillator. |
| XCIN, XCOUT | Sub-clock I/O pin | I/O | Input high or low, or leave open. |
| MD | Operating mode control | Input | Connect the VSS pin, or connect the VSS pin via a resistor (pull down). |
| P14/UB# | Operating mode control | Input | Connect the VSS pin, or connect the VSS pin via a resistor (pull down). For pull down, connect a resistor of up to 2.5 kΩ. |
| P35/UPSEL | USB power mode control | Input | Connect the VSS pin, or connect the VSS pin via a resistor (pull down). |
| RES# | Reset input | Input | Reset pin. Connect to the reset circuit. |
| USB0_DP | USB on-chip transceiver D+ I/O pin | I/O | Connect the circuit described in section 25, USB 2.0 Host/ Function Module (USBc). |
| USB0_DM | USB on-chip transceiver D- I/O pin | I/O | Connect the circuit described in section 25, USB 2.0 Host/ Function Module (USBc). |
| P16/USB0_VBUS | USB cable connection monitor pin | Input | Connect the circuit described in section 25, USB 2.0 Host/ Function Module (USBc). |
| P03, P05<br>P15, P17<br>P26, P27<br>P30, P31, P32<br>P40, P41, P42, P43, P44, P46<br>P54, P55<br>PA0, PA1, PA3, PA4, PA6<br>PB0, PB1, PB3, PB5, PB6, PB7<br>PC2, PC3, PC4, PC5, PC6, PC7<br>PE0, PE1, PE2, PE3, PE4, PE5, PE6, PE7 | Input port | Input | Input high or low, or leave open. |




Note 1. Connect a 6 MHz, 8 MHz, 12 MHz, 16 MHz crystal, ceramic resonator or oscillator to the main clock oscillator.
Note 2. The pin is pulled up by an on-chip input pull-up resistor for a certain period after start-up in boot mode.
Note 3. UPSEL is assigned to the same pin as NMI. This is an example when the falling edge is selected by the NMI detection set bit.

**Figure 35.6 Example of Pin Connections in USB Interface Mode When Bus-Powered**

**Table 35.6 Pin Handling in USB Interface Mode in Bus-Powered Mode**

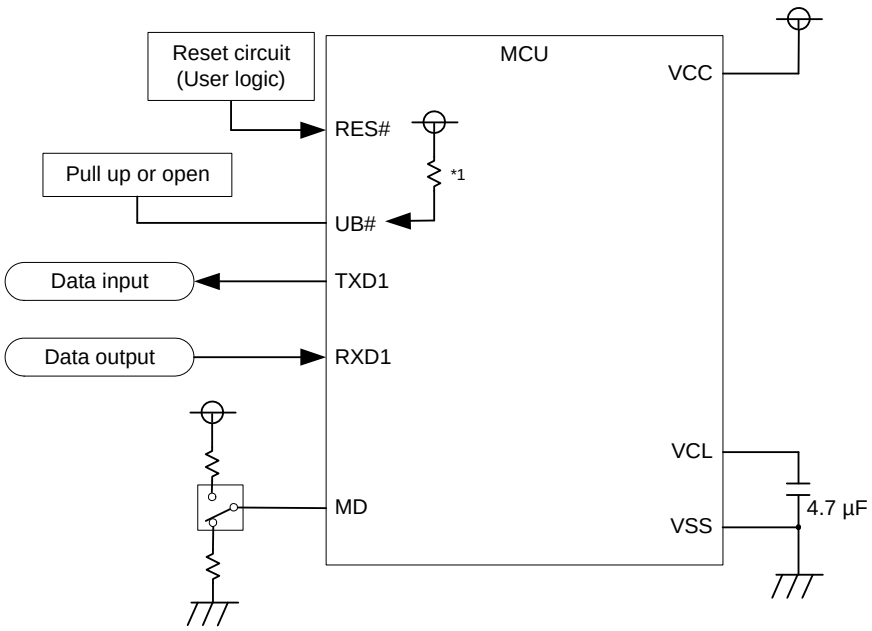
| Pin Name | Name | I/O | Function |
|---|---|---|---|
| VCC, VSS | Power supply input | Input | Input the guaranteed voltage for program/erase to the VCC pin. Input 0 V to the VSS pin. |
| VCC_USB, VSS_USB | USB Power supply input | Input | Connect the VCC_USB pin to the VCC pin. Connect the VSS_USB pin to the VSS pin. |
| AVCC0, AVSS0 | 12-bit A/D converter Power supply input | Input | Connect the AVCC0 pin to the VCC pin. Connect the AVSS0 pin to the VSS pin. |
| VREFH0, VREFL0 | 12-bit A/D converter reference power supply input | Input | Connect the VREFH0 pin to the VCC pin. Connect the VREFL0 pin to the VSS pin. |
| VCL | Decoupling capacitor connect pin | — | Connect to the VSS pin via a decoupling capacitor (4.7 µF) for stabilizing the internal voltage. |
| XTAL, EXTAL | Main clock I/O pin | I/O | Connect a 6 MHz, 8MHz, 12 MHz, or 16 MHz crystal or ceramic resonator or oscillator. |
| XCIN, XCOUT | Sub-clock I/O pin | I/O | Input high or low, or leave open. |
| MD | Operating mode control | Input | Connect the VSS pin, or connect the VSS pin via a resistor (pull down). |
| P14/UB# | Operating mode control | Input | Connect the VSS pin, or connect the VSS pin via a resistor (pull down). For pull down, connect a resistor of up to 2.5 kΩ. |
| P35/UPSEL | USB power mode control | Input | Connect the VSS pin, or connect the VSS pin via a resistor (pull down). |
| RES# | Reset input | Input | Reset pin. Connect to the reset circuit. |
| USB0_DP | USB on-chip transceiver D+ I/O pin | I/O | Connect the circuit described in section 25, USB 2.0 Host/ Function Module (USBc). |
| USB0_DM | USB on-chip transceiver D- I/O pin | I/O | Connect the circuit described in section 25, USB 2.0 Host/ Function Module (USBc). |
| P16/USB0_VBUS | USB cable connection monitor pin | Input | Connect the USB0_VBUS pin to the VCC pin. |
| P03, P05<br>P15, P17<br>P26, P27<br>P30, P31, P32<br>P40, P41, P42, P43, P44, P46<br>P54, P55<br>PA0, PA1, PA3, PA4, PA6<br>PB0, PB1, PB3, PB5, PB6, PB7<br>PC2, PC3, PC4, PC5, PC6, PC7<br>PE0, PE1, PE2, PE3, PE4, PE5, PE6, PE7 | Input port | Input | Input high or low, or leave open. |

### 35.7.2 SCI Mode

The user area and the data area can be programmed and erased using asynchronous serial communication in SCI mode. Contact the manufacturer for details on the dedicated flash memory programmer (SCI programmer).

#### 35.7.2.1 System Configuration in SCI Mode

SCI1 is used to communicate with the programmer in SCI mode.
Prepare tools for transmitting/receiving control commands and status via asynchronous serial communication, and data for programming in the dedicated flash memory programmer. Figure 35.7 shows the Example of Pin Connections in SCI Mode. Table 35.7 lists Pin Handling in SCI Mode.
Pins in Figure 35.7 show only those pins associated with SCI mode. Operations are not guaranteed in all systems.




Note: • A crystal or ceramic resonator is not required because the high-speed on-chip oscillator (HOCO) is used.
Note 1. The pin is pulled up by the on-chip pull-up resistor after start-up in boot mode.

**Figure 35.7 Example of Pin Connections in SCI Mode**

**Table 35.7 Pin Handling in SCI Mode**

| Pin Name | Name | I/O | Function |
|---|---|---|---|
| VCC, VSS | Power supply input | Input | Input the guaranteed voltage for program/erase to the VCC pin. Input 0 V to the VSS pin. |
| VCC_USB, VSS_USB | USB Power supply input | Input | Connect the VCC_USB pin to the VCC pin. Connect the VSS_USB pin to the VSS pin. |
| AVCC0, AVSS0 | 12-bit A/D converter power supply input | Input | Connect the AVCC0 pin to the VCC pin. Connect the AVSS0 pin to the VSS pin. |
| VREFH0, VREFL0 | 12-bit A/D converter reference power supply input | Input | Connect the VREFH0 pin to the VCC pin. Connect the VREFL0 pin to the VSS pin. |
| VCL | Decoupling capacitor connect pin | — | Connect to the VSS pin via a decoupling capacitor (4.7 µF) for stabilizing the internal voltage. |
| XTAL, EXTAL | Main clock I/O pin | I/O | Input high or low, or leave open. |
| XCIN, XCOUT | Sub-clock I/O pin | I/O | Input high or low, or leave open. |
| MD | Operating mode control | Input | Connect the VSS pin, or connect the VSS pin via a resistor (pull down). |
| P14/UB# | Operating mode control | Input | Connect the VCC pin via a resistor (pull up), or leave open. |
| RES# | Reset input | Input | Reset pin. Connect to the reset circuit. |
| P15/RXD1 | Operating mode control | Input | Input pin for Input serial data |
| P16/TXD1 | Operating mode control | Input | Output pin for output serial data |
| USB0_DP | USB on-chip transceiver D+ I/O pin | I/O | Leave the pin open. |
| USB0_DM | USB on-chip transceiver D- I/O pin | I/O | Leave the pin open. |
| P03, P05<br>P17<br>P26, P27<br>P30, P31, P32, P35<br>P40, P41, P42, P43, P44, P46<br>P54, P55<br>PA0, PA1, PA3, PA4, PA6<br>PB0, PB1, PB3, PB5, PB6, PB7<br>PC2, PC3, PC4, PC5, PC6, PC7<br>PE0, PE1, PE2, PE3, PE4, PE5, PE6, PE7 | Input port | Input | Input high or low, or leave the pin open. |

As shown in Figure 35.8, set the format to 8-bit data, 1 stop bit, no parity, and LSB first to communicate with the dedicated flash memory programmer.

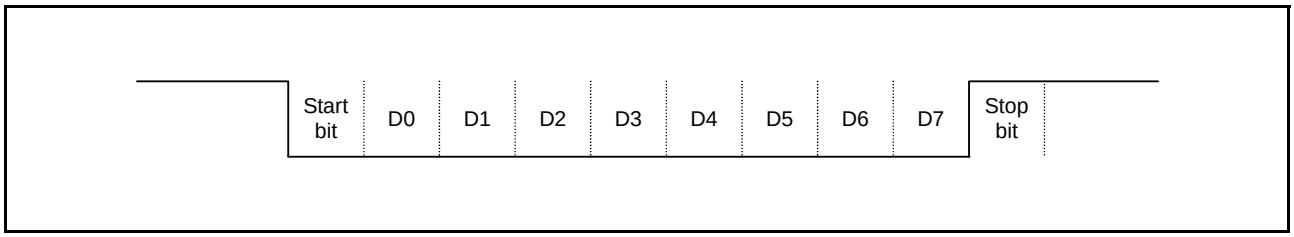



**Figure 35.8 Communication Format**

Communication with the programmer is performed at 9,600 bps or 19,200 bps. The communication bit rate can be changed after the MCU is connected with the programmer.
Table 35.8 lists the maximum communication bit rates for communication in SCI mode.

**Table 35.8 Conditions for Communication**

| Operating Voltage | Maximum Communication Bit Rate |
|---|---|
| 1.8 V or higher, and lower than 3.0 V | 500 kbps |
| 3.0 V or higher, and 3.6 V or lower | 2 Mbps |

### 35.7.2.2 Starting Up in SCI Mode

To start up in SCI mode, release the reset (drive the RES# pin high from low) while the MD pin is low and the UB# pin is high or open. After starting up in SCI mode, wait at least 400 ms holding the RES# pin high until communication is enabled in SCI mode.
Figure 35.9 shows the pin states until communication is enabled in SCI mode. Use resets according to the range described in section 36.3.2, Reset Timing.

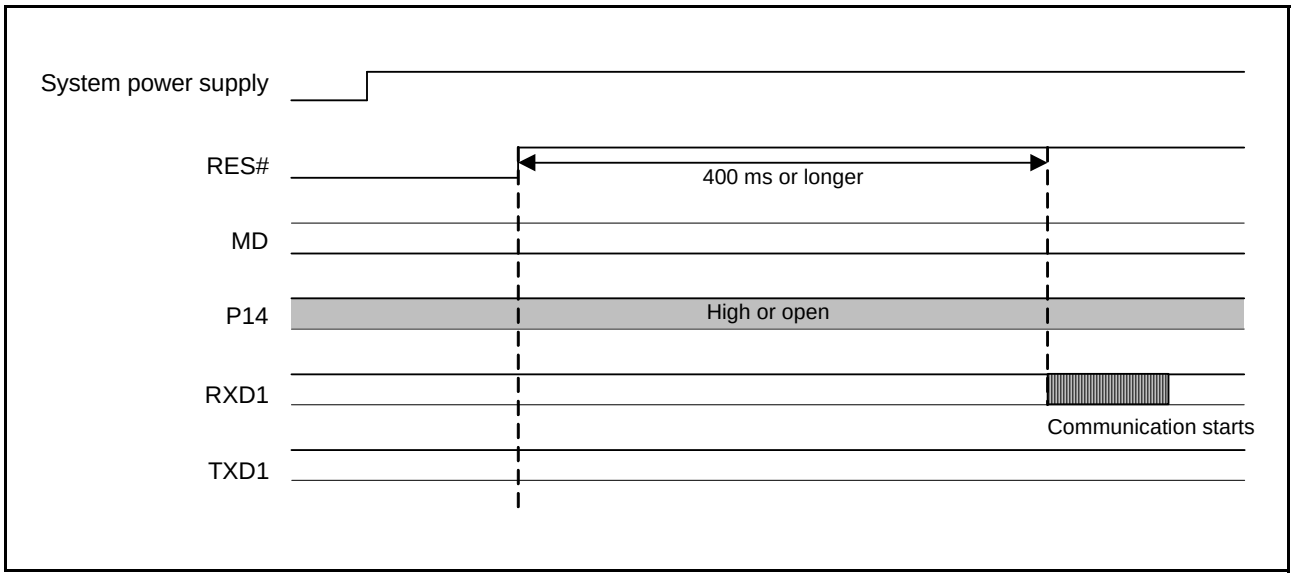



**Figure 35.9 Pin States until Communication Becomes Possible in SCI Mode**

## 35.8 Flash Memory Access Disable Function

The flash memory access disable function disables reading and programming of the flash memory. The boot mode ID code protection is for boot mode, and the on-chip debugging emulator ID code protection is for the on-chip debugging emulator. Details are below.

### 35.8.1 ID Code

ID codes are used for boot mode ID code protection in boot mode, and on-chip debugging emulator ID code protection in the on-chip debugging emulator.
ID codes consist of the control code and ID code 1 to ID code 15. Set ID codes in 32-bit units. Figure 35.10 shows the ID Code Configuration.

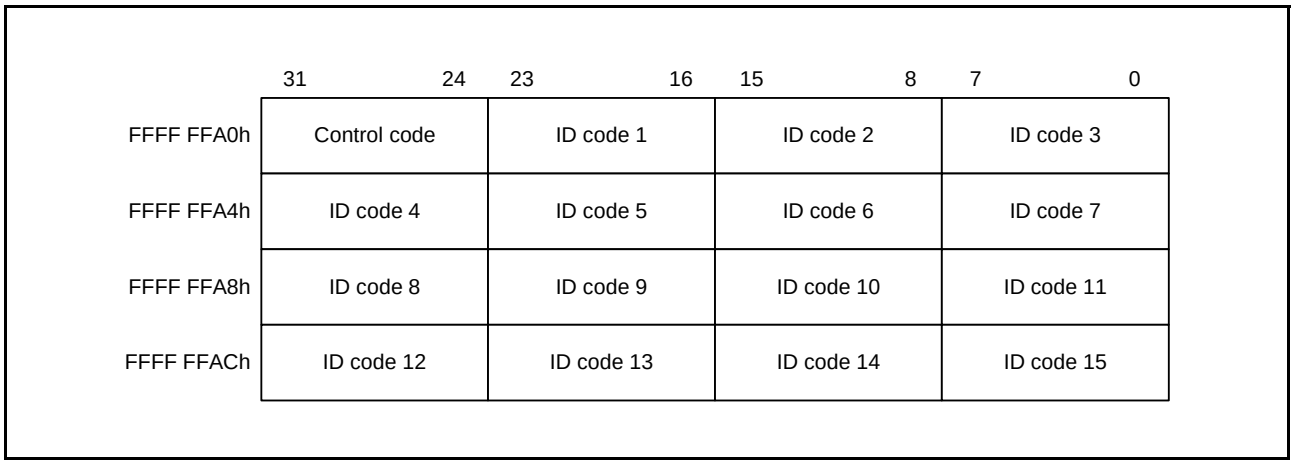



**Figure 35.10 ID Code Configuration**

The following shows a program example for setting ID codes
This is an example when setting the control code to 45h and setting ID codes to 01h, 02h, 03h, 04h, 05h, 06h, 07h, 08h, 0Ah, 0Bh, 0Ch, 0Dh, 0Eh, and 0Fh (from the ID code 1 field to the ID code 15 field).

C language:

```
#pragma address ID_CODE=0xFFFFFFA0
const unsigned long ID_CODE [4] = { 0x45010203, 0x04050607,0x08090A0B, 0x0C0D0E0F };
```

Assembly language:

```
.SECTION ID_CODE,CODE
.ORG 0FFFFFFA0h
.LWORD 45010203h
.LWORD 04050607h
.LWORD 08090A0Bh
.LWORD 0C0D0E0Fh
```

### 35.8.1.1 Boot Mode ID Code Protection

Boot mode ID code protection disables reading and programming of the user area and data area.
When the control code indicates that the boot mode ID code protection is disabled while the user area and data area are blank, the user area and data area can be read and programmed.
When the control code indicates that the boot mode ID code protection is disabled while the user area and data area have data, the MCU enters the erase ready state so the user area and data area can be erased. After all blocks in the user area and data area are erased in the erase ready state, the user area and data area can be read and programmed.
When the control code indicates that boot mode ID code protection is enabled, the MCU compares ID codes sent from the programmer with the control code and ID code 1 to ID code 15 in the user area. According to the comparison result, reading and programming the user area and data area are disabled.

(1) Control Code

The control code determines whether protection is enabled or disabled and the method of authentication with the programmer. Table 35.9 lists the protection specifications and Figure 35.11 shows the protection authentication flow.

**Table 35.9 Boot Mode ID Code Protection Specifications**

| Control Code | ID Code 1 to ID Code 15 | Protection | ID Code Matching Result | Content of User Area and Data Area | Operation |
|---|---|---|---|---|---|
| 45h | Any desired value | Enabled (authentication method 1) | Matched | — | Exit the boot mode ID code authentication state and enter the program/erase state. |
| | | | Not matched | — | Enter the boot mode ID code authentication state again. |
| | | | Not matched three times consecutively | Not blank | Enter the erase ready state for erasing the user area and data area. |
| 52h | Any value other than 50h, 72h, 6Fh, 74h, 65h, 63h, 74h, FFh, ..., and FFh | Enabled (authentication method 2) | Matched | — | Exit the boot mode ID code authentication state and enter the program/erase state. |
| | | | Not matched | — | Enter the boot mode ID code authentication state. |
| | 50h, 72h, 6Fh, 74h, 65h, 63h, 74h, FFh, ..., and FFh | Enabled (authentication method 3) | Not matched | — | Determine that the ID codes do not match regardless of the ID codes sent from the programmer, and enter the boot mode ID code authentication state again. |
| Other than above | Any desired value | Disabled | N/A | Blank | Enter the program/erase status |
| | | | | Not blank | Enter the erase ready state for erasing the user area and data area. |

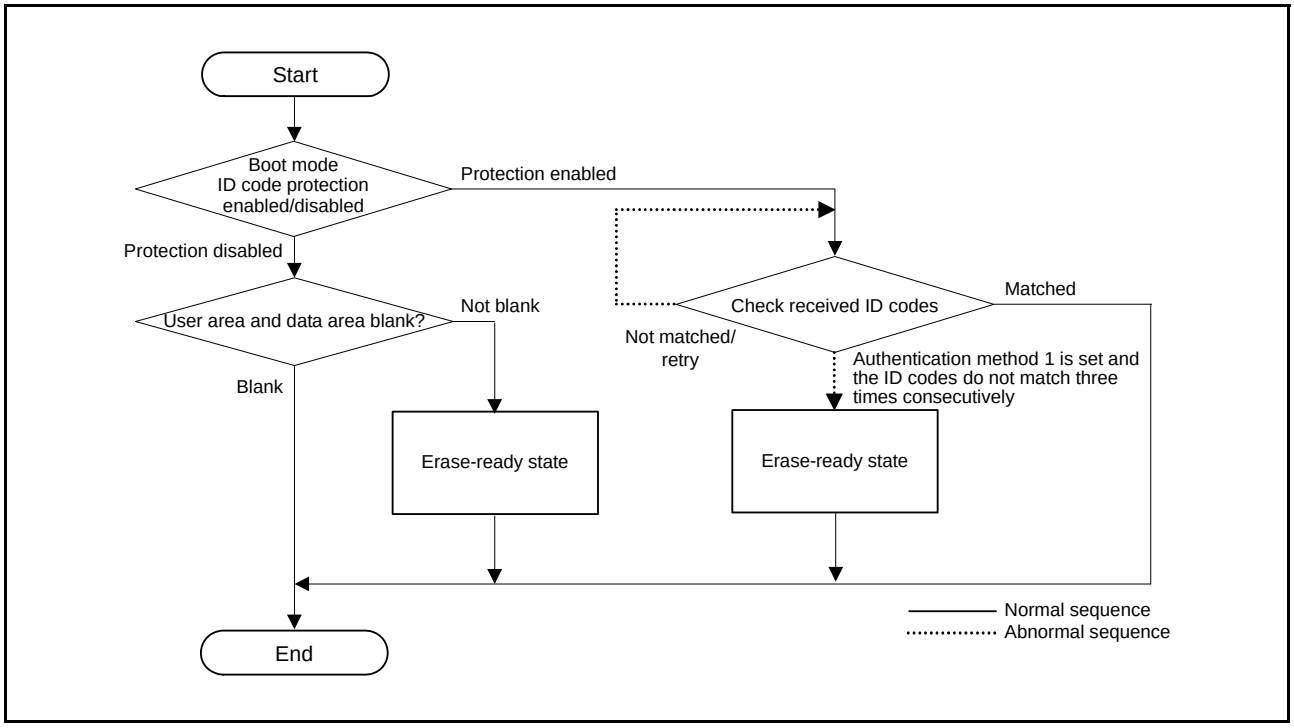



**Figure 35.11 Authentication for Boot Mode ID Code Protection**

(2) ID Code 1 to ID Code 15

ID code 1 to ID code 15 can be set to any desired value.

However, when using authentication method 3, the ID codes must be set to 50h, 72h, 6Fh, 74h, 65h, 63h, 74h, FFh, FFh, FFh, FFh, FFh, FFh, FFh, and FFh (from the ID code 1 field to the ID code 15 field).

### 35.8.1.2 On-Chip Debugging Emulator ID Code Protection

On-chip debugging emulator ID code protection enables or disables connection with the on-chip debugging emulator. When the on-chip debugging emulator ID code protection is disabled, connection with the on-chip debugging emulator is enabled. When ID codes sent from the on-chip debugging emulator and ID codes in the user area match while on-chip debugging emulator ID code protection is enabled, connection with the on-chip debugging emulator is also enabled. The ID code configuration shown in Figure 35.10 is used for the on-chip debugging emulator ID code protection. Table 35.10 lists the protection specifications.

**Table 35.10 On-Chip Debugging Emulator ID Code Protection Specifications**

| ID Code | Protection | ID Code Matching Result | Operation |
|---|---|---|---|
| FFh,…, and FFh (all FFh) | Disabled | N/A | Connect to the on-chip debugging emulator. (Connection with the on-chip debugging emulator is enabled.) |
| 52h, 50h, 72h, 6Fh, 74h, 65h, 63h, and 74h + any 8 bytes | Enabled | Not matched | Determine that the ID codes do not match regardless of the ID codes sent from the on-chip debugging emulator, and enter the ID code wait state again. (Connection with the on-chip debugging emulator is disabled.) |
| Other than above | Enabled | Matched | Complete ID code determination and connect to the on-chip debugging emulator. (Connection with the on-chip debugging emulator is enabled.) |
| | | Not matched | Enter the ID code wait state again. |

## 35.9 Communication Protocol

This section describes the protocol used in boot mode. When developing a programmer, control with this communication protocol.

### 35.9.1 State Transition in SCI Mode

Figure 35.12 shows the SCI Mode State Transition. Descriptions for numbers in parenthesis are on the following page.

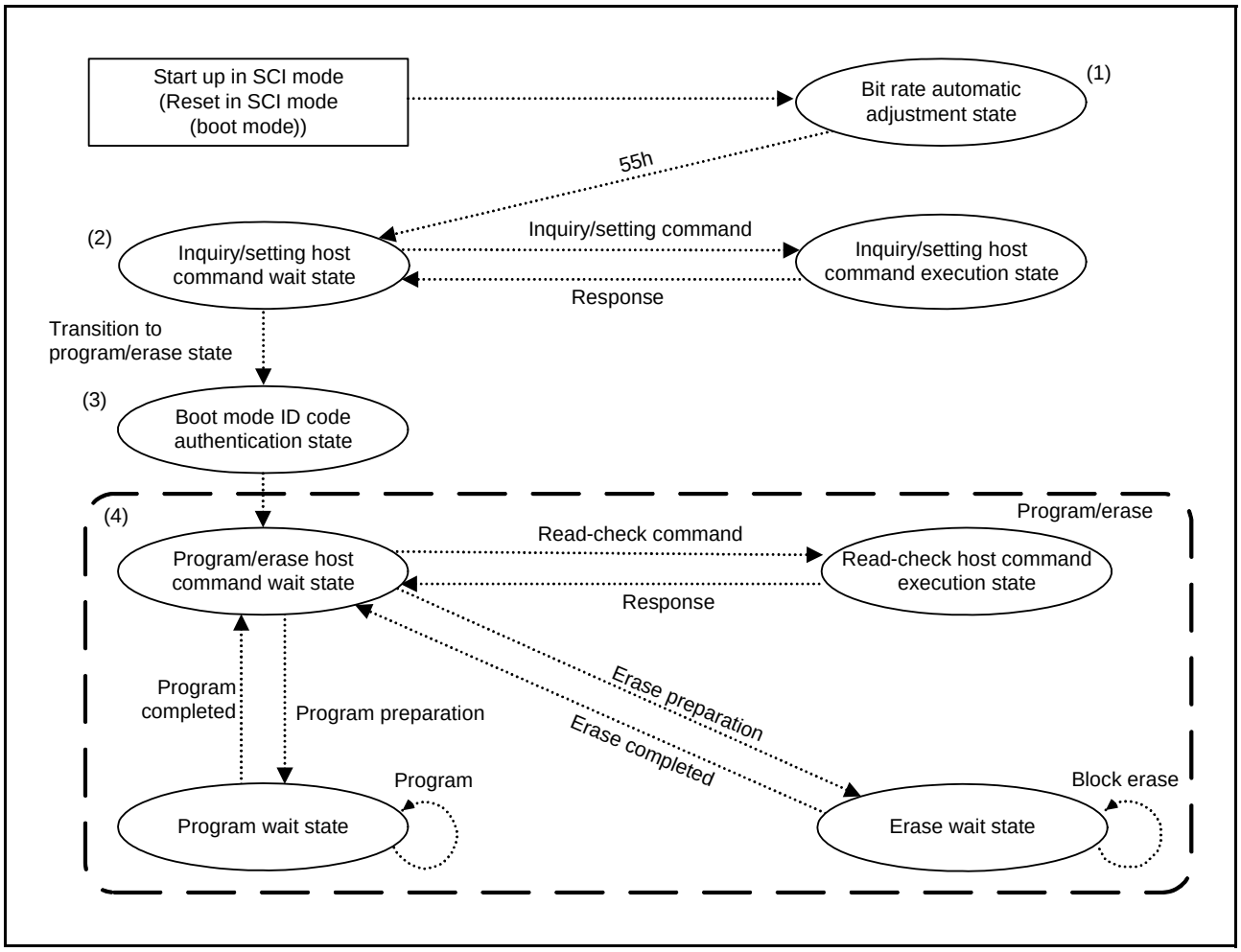



**Figure 35.12 SCI Mode State Transition**

### (1) Bit rate automatic adjustment state

In this state, the bit rate is automatically adjusted for communication with the host.
When the bit rate adjustment is completed, the MCU sends 00h to the host. After that, when the MCU receives 55h sent from the host, the MCU sends E6h to the host, and the MCU enters the inquiry/setting host command wait state.
The host must not send data until 400 ms elapse after a reset of the MCU is released.

### (2) Inquiry/setting host command wait state

In this state, the host can make inquiries for the MCU information including area configuration, size, and addresses, and select a device and bit rate.
When the host sends a program/erase state transition command, the MCU enters the boot mode ID code authentication state.
Refer to section 35.9.4, Inquiry Commands and section 35.9.5, Setting Commands for details on inquiry/setting commands.

### (3) Boot mode ID code authentication state

In this state, the MCU compares ID codes in boot mode ID code protection.
When the user area and data area are blank while boot mode ID code protection is disabled, the MCU enters the program/erase state. When the user area and data area are not blank while boot mode ID code protection is disabled, the MCU enters the erase ready state to erase the user area and data area. When all blocks in the user area and data area are erased in the erase ready state, the MCU enters the program/erase state. When boot mode ID codes do not match, the MCU remains in the boot mode ID code authentication state.
Refer to section 35.8.1.1, Boot Mode ID Code Protection for details on boot mode ID code protection. Refer to section 35.9.7, ID Code Authentication Command for details on the ID code authentication command.

### (4) Program/erase state

In this state, the MCU executes program/erase or read-check commands according to commands sent from the host.
Refer to section 35.9.8, Program/Erase Commands for details on program/erase commands. Refer to section 35.9.9, Read-Check Commands for details on read-check commands.

## 35.9.2 Command and Response Configuration

The communication protocol is composed of a “Command” sent from the host to the MCU and a “Response” sent from the MCU to the host. Commands include 1-byte commands and multiple-byte commands. Responses include 1-byte responses, multiple-byte responses, and error responses.
A multiple-byte command and multiple-byte response have “Size” for informing the number of transmit/receive data bytes and “SUM” for detecting communication errors.
“Size” indicates the number of transmit/receive data bytes excluding Command (the first byte), Size, SUM.
“SUM” indicates byte data that is calculated so the total bytes of Command or Response becomes 00h.

When the host sends an undefined command, the MCU sends a command error as a response. The contents of the response is shown below. The command in the error response stores the first byte of the command sent from the MCU.

| | | |
|---|---|---|
| Error response | 80h | Command |

### 35.9.3 Boot Mode Status Inquiry

This command is used to check the current status and the previous error of the boot program. The MCU returns a code from Table 35.11 and Table 35.12 as the current status and the previous error.
The boot mode status inquiry command can be used in the inquiry/setting host command wait state and program/erase state.

| | | | | |
|---|---|---|---|---|
| Command | 4Fh | | | |
| Response | 5Fh | Size | Status | Error |

Size (1 byte): Total bytes of “Status” and “Error” (the value is always 2)
Status (1 byte): MCU status (see Table 35.11)
Error (1 byte): Information about the error occurred in the MCU (see Table 35.12)

**Table 35.11 Information Regarding the States**

| Code | Status *1 | Description |
|---|---|---|
| 11h | Inquiry/setting host command wait state | Device selection wait state |
| 12h/ 13h | | Clock mode selection wait state |
| 1Fh | | Program/erase state transition command wait state |
| 3Fh | Program/erase state | Program/erase host command wait state |
| 4Fh | | Program wait state |
| 5Fh | | Erase wait state |

Note 1. Refer to Figure 35.12 for details on the states.

**Table 35.12 Error Information**

| Code | Description |
|---|---|
| 00h | No error |
| 11h | SUM error |
| 21h | Device code error |
| 24h | Bit rate selection error |
| 29h | Block address error |
| 2Ah | Address error |
| 2Bh | Data length error |
| 51h | Erase error |
| 52h | Not blank |
| 53h | Program error |
| 80h | Command error |
| FFh | Bit rate automatic adjustment error |

### 35.9.4 Inquiry Commands

Inquiry commands are used to obtain necessary information for sending setting commands, program/erase commands, and read-check commands. Table 35.13 lists the inquiry commands. These commands can only be used in the inquiry/setting host command wait state.

**Table 35.13 Inquiry Commands**

| Command | Description |
|---|---|
| Supported device inquiry | Inquiry for the device code and series name |
| Data area availability inquiry | Inquiry for the availability of the data area |
| User area information inquiry | Inquiry for the number of user areas, and the start and end addresses of the user area |
| Data area information inquiry | Inquiry for the number of data areas, and the start and end addresses of the data area |
| Block information inquiry | Inquiry for the start and end addresses of the user areas and data areas, the block size, and the number of blocks |

#### 35.9.4.1 Supported Device Inquiry

When the host sends this command, the MCU sends information about a device for little endian data and a device for big endian data in this order.

| | | | | |
|---|---|---|---|---|
| Command | 20h | | | |
| Response | 30h | Size | Number of devices | |
| | Number of characters | Device code for little endian | | Series name for little endian |
| | Number of characters | Device code for big endian | | Series name for big endian |
| | SUM | | | |

Size (1 byte): Total bytes of Number of Devices, Characters, Device code, and Series name
Number of devices (1 byte): Number of devices supported by the boot program
Number of characters (1 byte): Number of characters for the device code and device name
Device code (4 bytes): Identification code indicating the device
Series name (n bytes): ASCII code of the series name of the supported device
SUM (1 byte): Value that is calculated so the sum of response data is 00h

#### 35.9.4.2 Data Area Availability Inquiry

When the host sends this command, the MCU sends data indicating that the data area is available and area protection can be used.

| | | | | |
|---|---|---|---|---|
| Command | 2Ah | | | |
| Response | 3Ah | Size | Availability | SUM |

Size (1 byte): Number of characters of Availability (the value is always 01h)
Availability (1 byte): Availability of the data area (the value is always 19h)
19h represents the data area is available and area protection can be used.
SUM (1 byte): Value that is calculated so the sum of response data is 00h (the value is always ACh)

### 35.9.4.3 User Area Information Inquiry

When the host sends this command, the MCU sends the number of user areas and addresses.

| | | | |
|---|---|---|---|
| Command | 25h | | |
| Response | 35h | Size | Number of areas |
| | Area start address | | |
| | Area end address | | |
| | SUM | | |

Size (1 byte): Total bytes of Number of areas, Area start address, and Area end address (the value is always 09h)
Number of areas (1 byte): Number of user areas (the value is always 01h)
Area start address (4 bytes): Start address of the user area
Area end address (4 bytes): End address of the user area
SUM (1 byte): Value that is calculated so the sum of the response data is 00h

### 35.9.4.4 Data Area Information Inquiry

When the host sends this command, the MCU sends the number of data areas and addresses.

| | | | |
|---|---|---|---|
| Command | 2Bh | | |
| Response | 3Bh | Size | Number of areas |
| | Area start address | | |
| | Area end address | | |
| | SUM | | |

Size (1 byte): Total bytes of data of Number of areas, Area start address, and Area end address (the value is always 09h)
Number of areas (1 byte): Number of areas in the data area (the value is always 01h)
Area start address (4 bytes): Start address of the data area (the value is always 0010 0000h)
Area end address (4 bytes): End address of the data area (the value is always 0010 1FFFh)
SUM (1 byte): Value that is calculated so the sum of the response data is 00h (the value is always 7Dh)

### 35.9.4.5 Block Information Inquiry

When the host sends this command, the MCU sends the start address, the size of one block, and the number of blocks for the user area and data area.

| | | | |
|---|---|---|---|
| Command | 26h | | |

| | | | |
|---|---|---|---|
| Response | 36h | Size | DDh |
| | Start address of the user area | | |
| | Block size of one block for the user area | | |
| | Number of blocks of the user area | | |
| | Start address of the data area | | |
| | Block size of one block for the data area | | |
| | Number of blocks of the data area | | |
| | SUM | | |

Size (2 bytes): Total bytes of data from DDh to Number of blocks of the data area (the value is always 00 19h)
Start address of the user area (4 bytes): Start address of the user area
Block size of one block for the user area (4 bytes): Memory size of one block (the value is always 00 00 04 00h)
Number of blocks of the user area (4 bytes): Number of blocks in the user area
Start address of the data area (4 bytes): Start address of the data area (the value is always 00 10 00 00h)
Block size of one block for the data area (4 bytes): Memory size of one block (the value is always 00 00 04 00h)
Number of blocks of the data area (4 bytes): Number of blocks in the data area (the value is always 00 00 04 00h)
SUM (1 byte): Value that is calculated so the sum of response data is 00h

### 35.9.5 Setting Commands

Setting commands are used to configure the settings necessary to execute program/erase commands in the MCU. Table 35.14 lists Setting Commands. These commands can be used only in the inquiry/setting host command wait state.

**Table 35.14 Setting Commands**

| Command | Functions |
|---|---|
| Device select | Select a device code |
| Operating frequency select | Change the bit rate for communication |
| Program/erase state transition | Enter the erase ready state |

#### 35.9.5.1 Device Select

To send a device code from the host, select a device code in the response to the support device inquiry command, and send it using the device select command. Select the device code corresponding to the endian of program data.
When the device is supported, the MCU sends a response. When the device is not supported or the received command is invalid, the MCU sends an error response.

| | | | | |
|---|---|---|---|---|
| Command | 10h | Size | Device code | SUM |

Size (1 byte): Number of characters of the device code (the value is always 04h)
Device code (4 bytes): Identification code indicating the device
(code in the response to the support device inquiry command)
SUM (1 byte): Value that is calculated so the sum of command data is 00h

| | |
|---|---|
| Response | 46h |

| | | |
|---|---|---|
| Error response | 90h | Error |

Error (1 byte): Error code
11h: SUM error
21h: Device code error

### 35.9.6 Operating Frequency Select

When the host sends a command to set the bit rate, select 16 MHz input clock and a bit rate with error of less than 4%. When the settings are supported, the MCU sends a response. When the settings are not supported or the transmitted command is invalid, the MCU sends an error response.

After a response is received, wait for a 1-bit period and then change the bit rate to the selected value from the host. When the MCU successfully receives communication confirmation data, the MCU sends a response. When the MCU fails to receive the communication confirmation data, the MCU sends an error response.

| Command | 3Fh | Size | Bit rate | Input frequency |
|---|---|---|---|---|
| | Number of Clocks | Multiplier 1 | Multiplier 2 | |
| | SUM | | | |

Size (1 byte): Total bytes of data of Bit rate, Input frequency, Number of clocks, and Multiplier (the value is always 07h)
Bit rate (2 bytes): New bit rate (e.g. 00C0h: 19,200 bps)
The value is calculated by dividing the bit rate by 100.
Input frequency (2 bytes): Input frequency of the MCU (the value is always 0640h: 16 MHz)
Number of clocks (1 byte): Types of clocks (the value is always 02h)
Multiplier 1 (1 byte): Multiplier of the input frequency for the system clock (ICLK) (the value is always 01h)
Multiplier 2 (1 byte): Multiplier of the input frequency for the peripheral clock (PCLK) (the value is always 01h)
SUM (1 byte): Value that is calculated so the sum of command data is 00h

| Response | 06h |
|---|---|

| Error response | BFh | Error |
|---|---|---|

Error (1 byte): Error code
11h: SUM error
24h: Bit rate selection error

| Communication Confirmation | 06h |
|---|---|

| Response | 06h |
|---|---|

| Error Response | FFh |
|---|---|

- Bit rate selection error
  A bit rate selection error occurs when the bit rate specified with the operating frequency select command cannot be set to a value with error of less than 4%. When the bit rate selected by the operating frequency select command is B, and the value of the SCI bit rate register (BRR) is N, the bit rate error is calculated by the following formula:

$$\text{Error (\%)} = \left\{ \left[ \frac{16 \times 10^6}{B \times 64 \times 2^{-1} \times (N + 1)} \right] - 1 \right\} \times 100$$

### 35.9.6.1 Program/Erase Status Transition

When the host sends this command, the MCU performs processing for the boot mode ID code protection.

When boot mode ID code protection is disabled while the user area and data area are blank, the MCU sends a response (06h) indicating that ID code protection is disabled and the MCU enters the program/erase state.
When boot mode ID code protection is disabled while the user area and data area are not blank, the MCU sends a response (56h) indicating that ID code protection is disabled and the MCU enters the erase ready state.
When boot mode ID code protection is enabled, the MCU sends a response (16h) indicating that ID code protection is enabled.

| Command | 40h |
|---|---|

| Response | ACK |
|---|---|

ACK (1 byte): ACK code
06h: ID code protection is disabled. The user area and data area are blank. *1
56h: ID code protection is disabled.
16h: ID code protection is enabled.

Note 1. Erase the block that is programmed before sending a program command.

### 35.9.7 ID Code Authentication Command

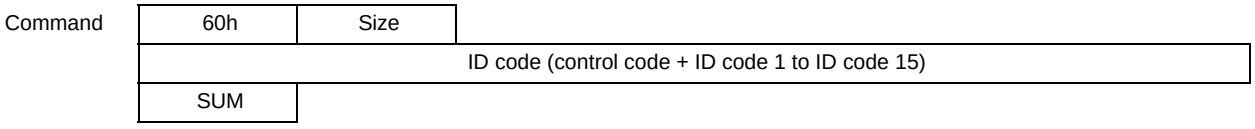
The ID code authentication command is used to send data from the host to compare with the control code and ID code 1 to ID code 15 on the ROM when the boot mode ID code protection is enabled.
Table 35.15 lists ID code authentication command. This command can be used only in the boot mode ID code authentication state.

**Table 35.15 ID Code Authentication Command**

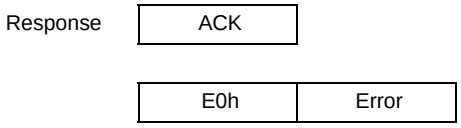
| Command | Function |
|---|---|
| ID code check | Compare data with the control code and ID code 1 to ID code 15 using boot mode ID code protection. |

#### 35.9.7.1 ID Code Check

Send the same data as the control code and ID codes 1 to ID code 15 in the user area from the host. When the control code and ID code 1 to ID code 15 match the boot mode ID codes, the MCU enters the program/erase state and sends a response (06h).
When they do not match three times consecutively in authentication method 1, the MCU enters the erase ready state and sends a response (56h). When they do not match or when the MCU fails to receive data, the MCU sends an error response.




Size (1 byte): Number of bytes of ID codes (the value is always 16)
ID code (16 bytes): Control code (1 byte) + ID code 1 to ID code 15 (15 bytes)
SUM (1 byte): Value that is calculated so the sum of the command data is 00h




ACK (1 byte): ACK code
06h: The MCU enters the program/erase state.
56h: The MCU enters the erase ready state.
Error (1 byte): Error code
11h: SUM error
61h: ID codes do not match

#### 35.9.7.2 Erase Ready

The erase ready is a part of the boot mode ID code protection to disable reading data stored in the user area and data area when protection is disabled while the user area and data area are not blank or when ID codes do not match three times consecutively in authentication method 1.
Only the erase preparation command and block erase command can be accepted in the erase ready state.
Table 35.16 lists ID Code Authentication Command.

**Table 35.16 Commands Used for Erase Ready**

| Command | Function |
|---|---|
| Erase preparation | Enter the erase wait state |
| Block erase | Erase the selected block, or enter the program/erase state (end of erase) |

Note: Refer to section 35.9.9, Read-Check Commands for details on the erase preparation command and block erase command.

### 35.9.8 Program/Erase Commands

Program/erase commands are used to program or erase the user area or data area based on the response to inquiry commands. Table 35.17 lists commands used in the program/erase command wait state, program wait state, and erase wait state. Table 35.18 lists commands that cannot be accepted in each state.
When a command that cannot be accepted is received in the state listed in Table 35.18, the MCU sends a command error response.

**Table 35.17 Program/Erase Commands**

| Command | Function |
|---|---|
| User/data area program preparation | Select the user area or data area to program, and enter the program wait state. |
| Program | Program the selected area, or enter the program erase state (end of program) |
| Erase preparation | Enter the erase wait state |
| Block erase | Erase the selected block, or enter the program erase state (end of erase) |

**Table 35.18 Commands That Cannot Be Accepted**

| State | Command That Cannot Be Accepted |
|---|---|
| Program/erase host command wait state | Commands other than the program command and block erase command |
| Program wait state | Commands other than the program command |
| Erase wait state | Commands other than the block erase command |

### 35.9.8.1 User/Data Area Program Preparation

When the host sends this command, the MCU recognizes that an instruction to prepare for the program command is issued from the host, enters the program wait state, where only the program command to the user area or data area can be accepted, and sends a response.

| | |
|---|---|
| Command | 43h |
| Response | 06h |

### 35.9.8.2 Program

Set the program address sent from the host aligned on a 256-byte boundary. When the data length is shorter than 256 bytes, the data cannot be programmed. Fill the gaps with FFh to send.
When the program from the selected address is successfully completed, the MCU sends a response. When an error occurs during a program operation, the MCU sends an error response.

When entering program/erase host command wait state, send 50h FFh FFh FFh FFh B4h from the host.
The MCU enters the program/erase host command wait state and sends a response.

| | | |
|---|---|---|
| Command | 50h | Program address |
| | Program data | |
| | SUM | |

Program address (4 bytes): Address for program destination
Address aligned on the program data length
Set FFFF FFFFh for end of program
Program data (n bytes): Program data (n = 256 in boot mode, 0 for end of program)
When the program is less than 256 bytes, set FFh for the missing data.
No data for the end of program
SUM (1 byte): Value that is calculated so the sum of command data is 00h.

| | | |
|---|---|---|
| Response | 06h | |
| Error response | D0h | Error |

Error (1 byte): Error code
11h: SUM error
2Ah: Address error (the address is not in the selected area.)
53h: Program error (the program cannot be written.)

### 35.9.8.3 Erase Preparation

When the host sends this command, the MCU recognizes that an instruction to prepare for the erase command is issued from the host, enters the erase wait state, where only the block erase command to the user area or data area can be accepted, and sends a response.

| | |
|---|---|
| Command | 48h |
| Response | 06h |

### 35.9.8.4 Block Erase

Send block addresses from the host based on the response to the block information inquiry command.
When the block selected in addresses is successfully erased, the MCU sends a response. When an error occurs during an erase operation, the MCU sends an error response.

When the MCU enters the program/erase host command wait state after the erase operation ends, send 59h 04h FFh FFh FFh FFh A7h from the host. The MCU enters the program/erase host command wait state and sends a response.

| | | |
|---|---|---|
| Command | 59h | Size |
| | Block start address | |
| Response | SUM | |

Size (1 byte): Total bytes of Block start address (the value is always 04h)
Block start address (4 bytes): Start address of the block that is erased
Set FFFF FFFFh for end of erase
SUM (1 byte): Value that is calculated so the sum of response data is 00h

| | |
|---|---|
| Response | 06h |

| | | |
|---|---|---|
| Error response | D9h | Error |

Error (1 byte): Error code
11h: SUM error
29h: Block address error (the block address is not correct)
51h: Erase error (the selected block cannot be erased)

### 35.9.9 Read-Check Commands

Read-check commands are used to read or check the user area or data area in the MCU based on the response to inquiry commands.
Table 35.19 lists read-check commands used in the program/erase command wait state.

**Table 35.19 Read-Check Commands**

| Command | Function |
|---|---|
| Memory read | Read data from the user area.<br>Read data from the data area. |
| User area checksum | Checksum of the user area |
| Data area checksum | Checksum of the data area |
| User area blank check | Check whether data is programmed in the user area. |
| Data area blank check | Check whether data is programmed in the data area. |
| Access window program | Set the access window |
| Access window read | Read the settings of the access window |

#### 35.9.9.1 Memory Read

For a read start address sent from the host, set an address within the range from the start address to the end address received in the response to the user area information inquiry command or the data area information inquiry command.
For a read size sent from the host, set a value so the sum of the read start address and the read size is within the range from the start address to the end address received in the response to the user area information inquiry command or the data area information inquiry command.
When the MCU performs a read successfully, the MCU sends data for the read size from the read start address. When the MCU fails to read the flash memory, the MCU sends an error response.

Command

| 52h | Size | Area |
|---|---|---|
| Read start address | | |
| Read size | | |
| SUM | | |

Size (1 byte): Total bytes for Read start address and Read size
Area (1 byte): Area that is read
01h: User area or data area
Read start address (4 bytes): Start address of the area that is read
Read size (4 bytes): Size of data that is read (in bytes)
SUM (1 byte): Value that is calculated so the sum of response data is 00h

Response

| 52h | Read size |
|---|---|
| Read data | |
| SUM | |

Read size (4 bytes): Size of Data that is read (in bytes)
Read data (read size): Data read from the selected address
SUM (1 byte): Value that is calculated so the sum of response data is 00h

| Error response | D2h | Error |
|---|---|---|

Error (1 byte): Error codes

11h: SUM error

2Ah: Address error

- A value other than 00h and 01h is set for area.
- The read start address is not in the selected area.

2Bh: Size error

- The read size is set to 00h.
- The read size exceeds the area size
- The address calculated from the read start address and read size is not in the selected area

### 35.9.9.2 User Area Checksum

When the host sends this command, the MCU adds data from the start address and the end address in bytes in the response to the user area information inquiry command, and sends the calculated result (checksum) as a response.

| Command | 4Bh |
|---|---|

| Response | 5Bh | Size |
|---|---|---|
| | Area checksum | |
| | SUM | |

Size (1 byte): Number of bytes for checksum of the area (the value is always 04h)
Area checksum (4 bytes): Checksum result of the user area
SUM (1 byte): Value that is calculated so the sum of response data is 00h

### 35.9.9.3 Data Area Checksum

When the host sends this command, the MCU adds data from the start address and the end address in bytes in the response to the data area information inquiry command, and sends the calculated result (checksum) as a response.

| Command | 61h |
|---|---|

| Response | 71h | Size |
|---|---|---|
| | Area checksum | |
| | SUM | |

Size (1 byte): Number of bytes for checksum of the area (the value is always 04h)
Area checksum (4 bytes): Checksum result of the data area
SUM (1 byte): Value that is calculated so the sum of response data is 00h

#### 35.9.9.4 User Area Blank Check

When the host sends this command, the MCU sends a response when there is no data in the area from the start address to the end address received in the response to the user area information inquiry command. When there is at least 1 byte of data, the MCU sends an error response.

| | | |
|---|---|---|
| Command | 4Dh | |
| Response | 06h | |
| Error response | CDh | Error |

Error (1 byte): Error code
52h: Not blank

#### 35.9.9.5 Data Area Blank Check

When the host sends this command, the MCU sends a response when there is no data in the area from the start address to the end address received in the response to the data area information inquiry command. When there is at least 1 byte of data, the MCU sends an error response.

| | | |
|---|---|---|
| Command | 62h | |
| Response | 06h | |
| Error response | E2h | Error |

Error (1 byte): Error code
52h: Not blank

### 35.9.9.6 Access Window Program

For the access window start address sent from the host, set the block start address of the user area. For the access window end address, set the block end address of the user area.
When the program of selected address is successfully completed, the MCU sends a response. When an error occurs during a program operation, the MCU sends an error response.
The access window settings can be overwritten because it is a part of protection.

| | | | | |
|---|---|---|---|---|
| Command | 74h | 05h | Access window | |
| | Access window start address LH | Access window start address HL | Access window end address LH | Access window end address HL |
| | SUM | | | |

Access window (1 byte): Select the access window or clear the access window settings
Set 00h to select the access window
Set FFh to clear the access window settings
Access window start address LH (1 byte): Start address of the access window (D15 to D8)
Set D15 to D8 of the block start address.
Set FFh to clear the access window settings
Access window start address HL (1 byte): Start address of the access window (D23 to D16)
Set D23 to D16 of the block start address.
Set FFh to clear the access window settings
Access window end address LH (1 byte): End address of the access window (D15 to D8)
Set D15 to D8 of the block end address.
Set FFh to clear the access window settings
Access window end address HL (1 byte): End address of the access window (D23 to D16)
Set D23 to D16 of the block end address.
Set FFh to clear the access window settings
SUM (1 byte): Value that is calculated so the sum of response data is 00h

| | | |
|---|---|---|
| Response | 06h | |
| Error response | F4h | Error |

Error (1 byte): Error code
11h: SUM error
2Ah: Address error (address is not in the selected area)
53h: Program error (program cannot be written)

### 35.9.9.7 Access Window Read

Send command 73h 01h FFh 8Dh from the host.
When the MCU successfully reads the access window settings, the MCU sends the access window start address and end address that the MCU read.
When the MCU fails to read the access window settings, the MCU sends an error response.

| | | | | |
|---|---|---|---|---|
| Command | 73h | 01h | FFh | 8Dh |

| | | | | |
|---|---|---|---|---|
| Response | 73h | 05h | | |
| | Access window start address LH | Access window start address HL | Access window end address LH | Access window end address HL |
| | FFh | | | |
| | SUM | | | |

Access window start address LH (1 byte): Start address of the access window (D15 to D8)
Access window start address HL (1 byte): Start address of the access window (D23 to D16)
Access window end address LH (1 byte): End address of the access window (D15 to D8)
Access window end address HL (1 byte): End address of the access window (D23 to D16)
SUM (1 byte): Value that is calculated so the sum of response data is 00h

| | | |
|---|---|---|
| Error response | F3h | Error |

Error (1 byte): Error code
11h: SUM error

### 35.9.10 Programmer Operation in SCI Mode

The following describes the procedure for the programmer to access the user area and data area in SCI mode.

1. Automatically adjust the bit rate
2. Receive the MCU information
3. Select the device and change the bit rate
4. Enter the program/erase state
5. Disable boot mode ID code protection
6. Perform erase ready processing
7. Erase the user area and data area
8. Program the user area and data area
9. Check data in the user area
10. Check data in the data area
11. Set the access window in the user area
12. Reset the MCU

When a timeout occurs or invalid response data is received, stop the operation and perform step 12 (reset the MCU).
If the necessary information has been already received, step 2 can be skipped.
Any step from 7 to 11 can be skipped, and their order can be changed.

Refer to section 35.9.10.1, Bit Rate Automatic Adjustment Procedure to section 35.9.10.11, Set the Access Window in the User Area for details on the procedure above. Refer to section 35.9.4, Inquiry Commands, section 35.9.5, Setting Commands, section 35.9.7, ID Code Authentication Command, section 35.9.8, Program/Erase Commands, and section 35.9.9, Read-Check Commands for details on commands.

### 35.9.10.1 Bit Rate Automatic Adjustment Procedure

The MCU measures the low width of data 00h to adjust the bit rate.

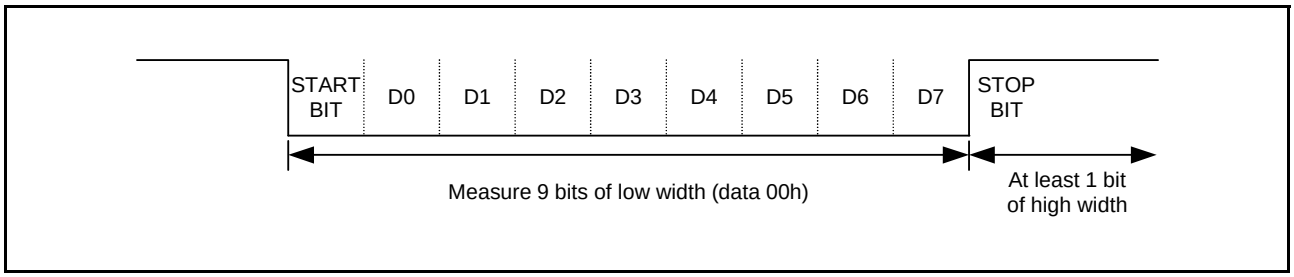



**Figure 35.13 Data Format for Bit Rate Automatic Adjustment**

After starting up in boot mode, wait for 400 ms and then send 00h to the MCU from the programmer. When the bit rate adjustment is completed, the MCU sends 00h to the programmer. When the programmer receives 00h, send 55h to the MCU from the programmer. When the programmer fails to receive 00h, restart the MCU in boot mode, and adjust the bit rate again. The programmer can send 00h to the MCU up to 30 times.
When the MCU receives 55h, the MCU sends E6h and enters the inquiry/setting command wait state. When the MCU fails to receive 55h, the MCU sends FFh. When the programmer receives FFh, restart the MCU in boot mode, and adjust the bit rate again.

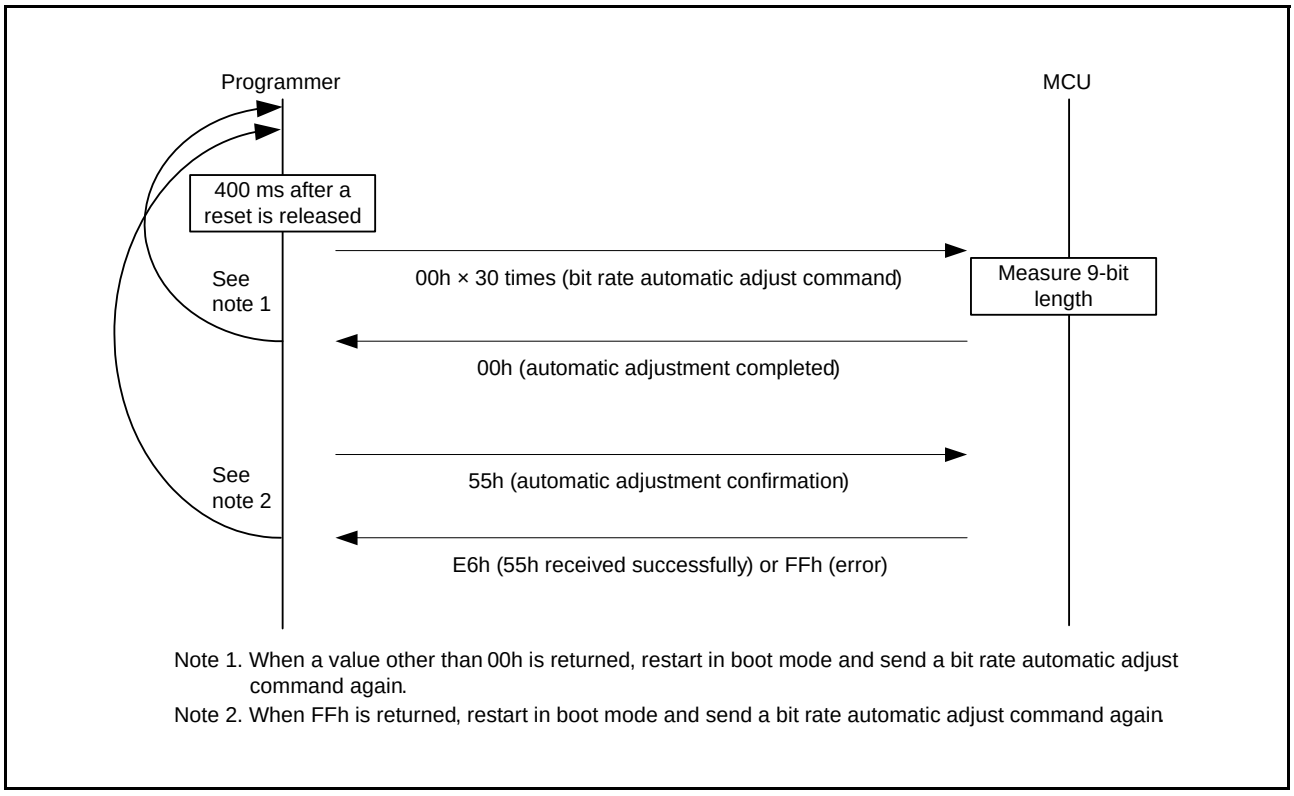



**Figure 35.14 Bit Rate Automatic Adjustment Procedure**

### 35.9.10.2 Procedure to Receive the MCU Information

Send inquiry commands, and receive the information necessary to send setting commands, program/erase commands, and read-check commands.

(1) Send a support device inquiry command (20h) to check which device to connect. The MCU returns the device code and series name.
(2) Send a data area availability inquiry command (2Ah) to check the availability of data area and area protection. The MCU returns the availability of data area and area protection.
(3) Send a user area information inquiry command (25h) to check the start and end addresses of the user area. The MCU returns the start and end addresses of the user area.
(4) Send a block information inquiry command (26h) to check the block configuration. The MCU returns the start address, the size of one block, and the number of blocks for the user area and data area.
(5) Send a data area information inquiry command (2Bh) to check the start and end addresses in the data area. The MCU returns the start and end addresses of the data area.

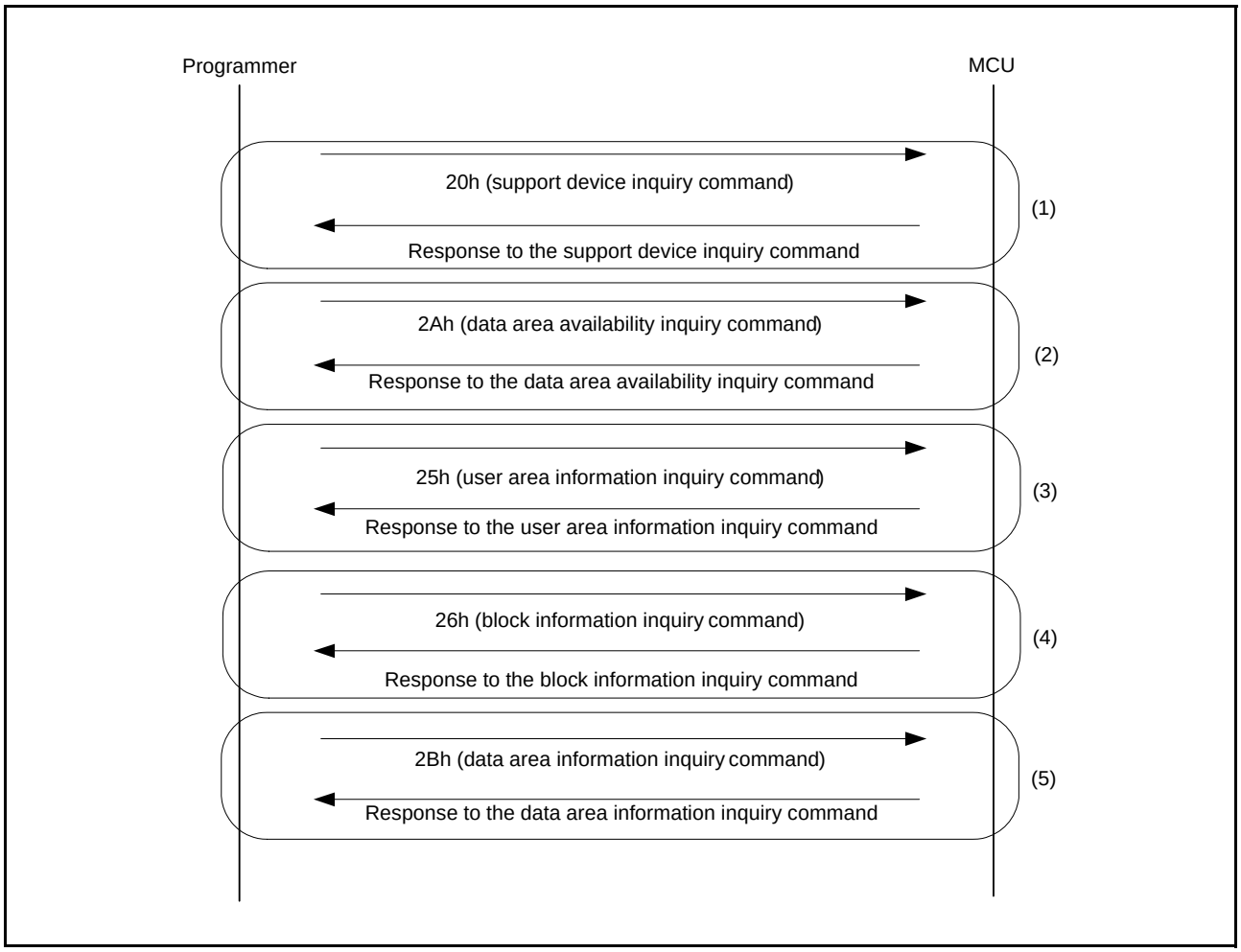



**Figure 35.15 Procedure to Send Inquiry Commands**

### 35.9.10.3 Procedure to Select the Device and Change the Bit Rate

Set the device to connect with the programmer and change the bit rate for communication.

(1) Send the device select command (10h) to select the device to connect with the programmer and the endian of data that is programmed. When the program data is little endian, select the same device code as that for little endian in the response to the support device inquiry command. When the program data is big endian, select the same device code as that for big endian in the response to the support device inquiry command. When the device is selected successfully, the MCU sends a response (06h). When the MCU fails to receive, the MCU sends an error response (90h).

(2) Send the operating frequency select command (3Fh) to change the bit rate for communication. When the bit rate is set successfully, the MCU sends a response (06h). When the bit rate cannot be changed, or when the MCU fails to receive, the MCU sends an error response (BFh).

(3) When the MCU receives a response (06h), the MCU waits for 1-bit period at the bit rate for sending the operating frequency select command, and then set the bit rate of the programmer to the changed value. After that, the MCU sends communication confirmation data (06h) at the changed bit rate. When the MCU receives the command successfully, the MCU sends a response (06h) of the communication confirmation data.

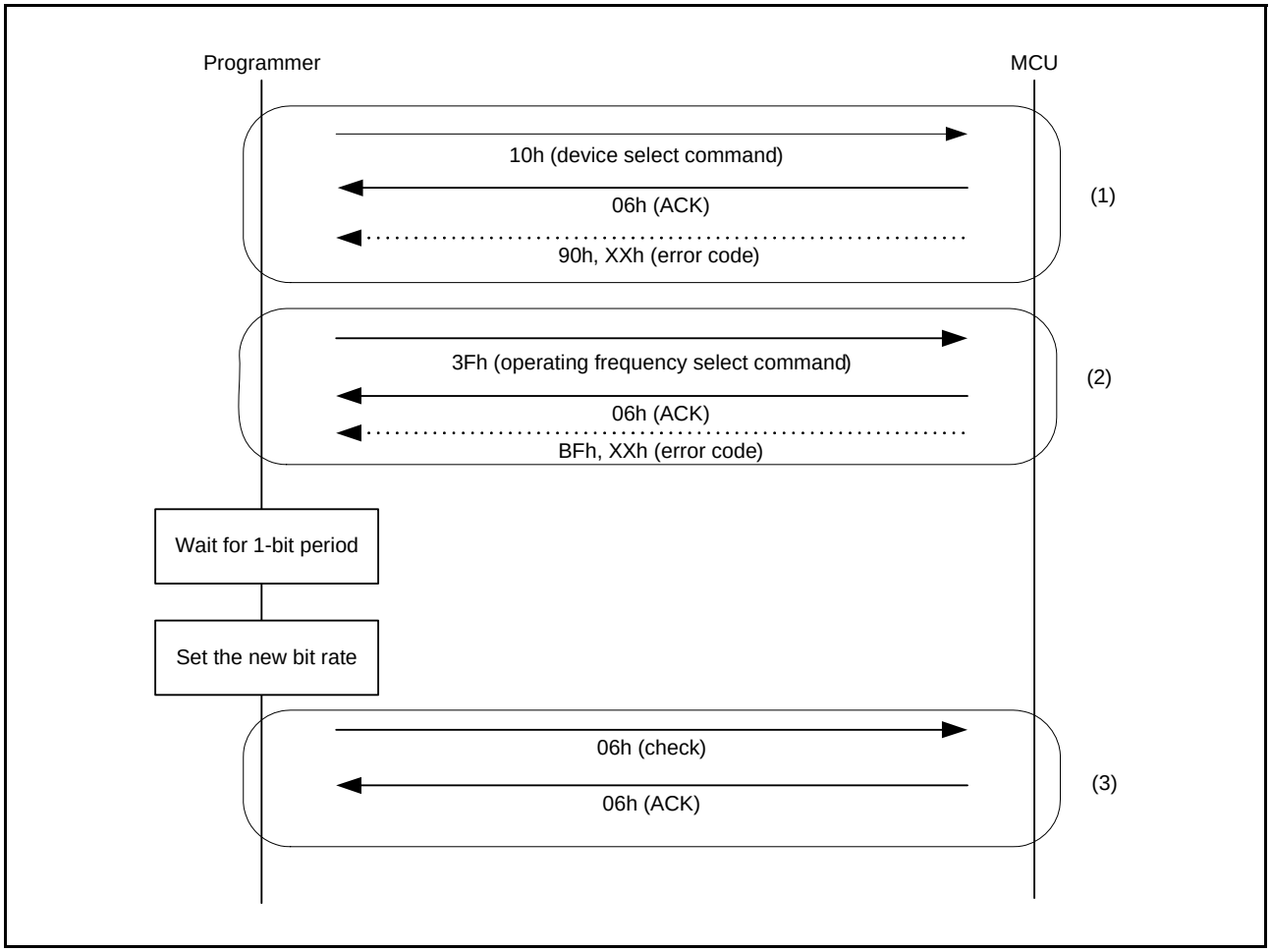



**Figure 35.16 Procedure to Select the Device and Change the Bit Rate**

### 35.9.10.4 Transition to the Program/Erase State

The MCU needs to enter the program/erase state to perform program/erase operations.
Send the program/erase status transition command (40h). The MCU responds according to ID codes and the status of the user area and data area.

(1) When boot mode ID code protection is disabled while the user area and data area are blank, the MCU sends a response (06h). After the MCU responds, the MCU is in the program/erase state and performs operations described in section 35.9.10.7, Erase the User Area and Data Area to section 35.9.10.11, Set the Access Window in the User Area. Perform an erase operation described in section 35.9.10.7, Erase the User Area and Data Area before programming.

(2) When the boot mode ID code protection is disabled while the user area and data area are not blank, the MCU sends a response (56h). After the MCU responds, the MCU is in the erase ready wait state and performs operation described in section 35.9.10.6, Erase Ready Operation.

(3) When the boot mode ID code protection is enabled, the MCU sends a response (16h). After the MCU responds, the MCU is in the ID code authentication wait state and performs operation described in section 35.9.10.5, Disable Boot Mode ID Code Protection.

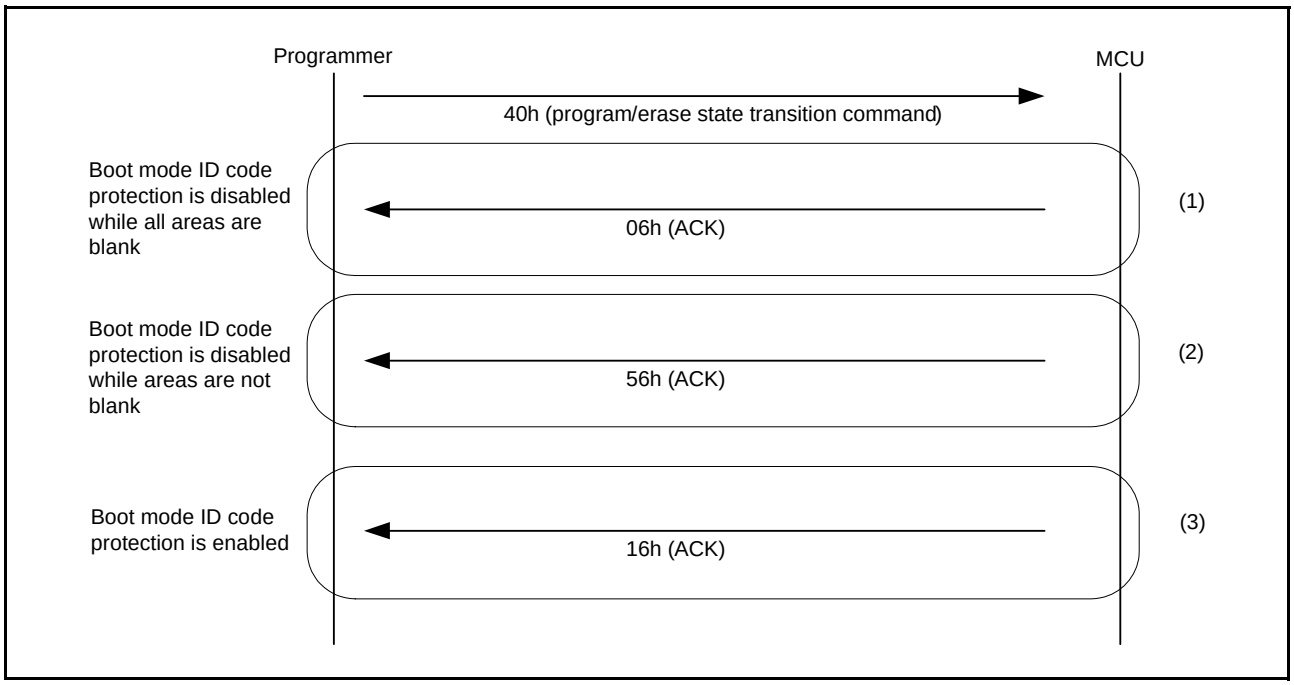



**Figure 35.17 Procedure to Enter the Program/Erase State**

### 35.9.10.5 Disable Boot Mode ID Code Protection

After the MCU is connected with the programmer, the boot mode ID code protection is enabled so program/read, and read-check operations cannot be performed. Disable this boot mode ID code protection.

Send the ID code check command (60h). When the MCU compares the received ID codes with ID codes in the user area and responds according to the comparison result.

(1) When ID codes match, the MCU sends a response (06h). After the MCU responds, the MCU is in the program/erase state and performs operations described in section 35.9.10.7, Erase the User Area and Data Area to section 35.9.10.11, Set the Access Window in the User Area. Data in the user area and data area are not erased. Perform an erase operation described in section 35.9.10.7, Erase the User Area and Data Area before programming.
When ID codes do not match, the MCU sends an error response (E0h). After the MCU responds, the MCU remains in the ID code authentication wait state.

(2) If ID codes do not match three times consecutively while the control code is 45h, the MCU sends a response (56h). After the MCU responds, the MCU is in the erase ready wait state and performs operation described in section 35.9.10.6, Erase Ready Operation.

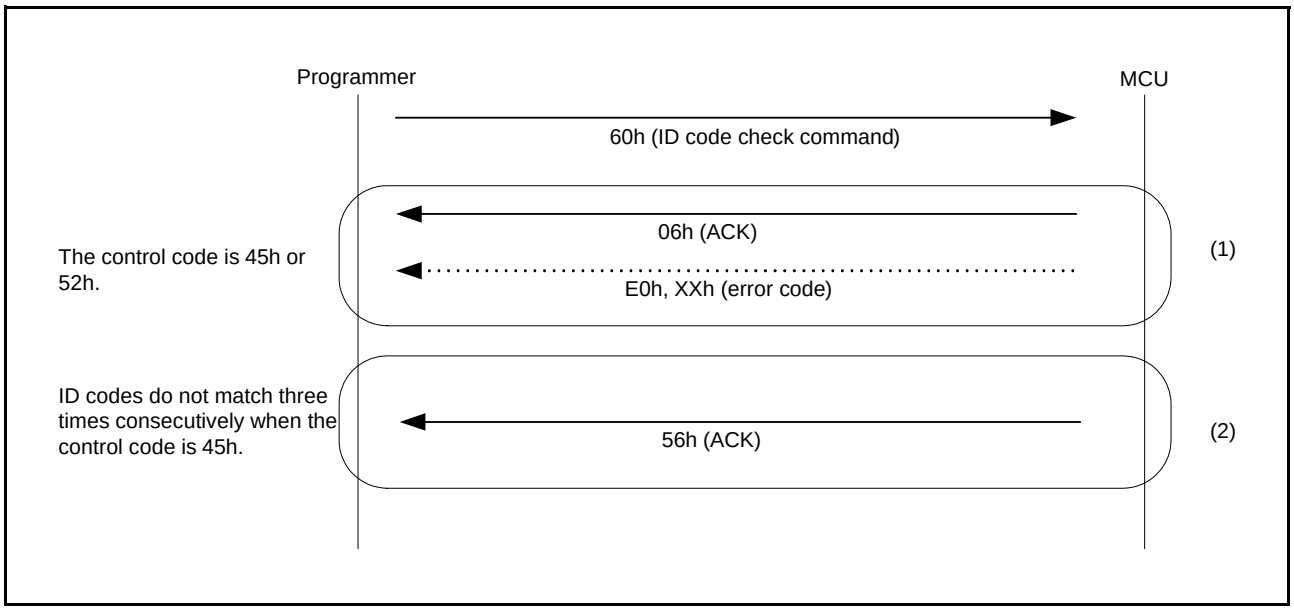



**Figure 35.18 Procedure to Check ID Codes**

### 35.9.10.6 Erase Ready Operation

Erase the user area and data area in the MCU.

(1) Send the erase preparation command (48h) to place the MCU in the erase wait state. The MCU enters the erase wait state and sends a response (06h).

(2) Send a block erase command (59h) to erase blocks in the MCU. When blocks are erased successfully, the MCU sends a response (06h). When the MCU fails to receive, the MCU sends an error response (D9h).
Send a block erase command repeatedly until block erase commands for all blocks are sent. When the operation ends before all the block erase commands are sent, a command error may occur even when a correct command is sent in the program/erase state.

(3) In order to place the MCU in the program/erase state, send a block erase command for end of erase (59h 04h FFh FFh FFh FFh A7h). The MCU enters the program/erase state and sends a response (06h).

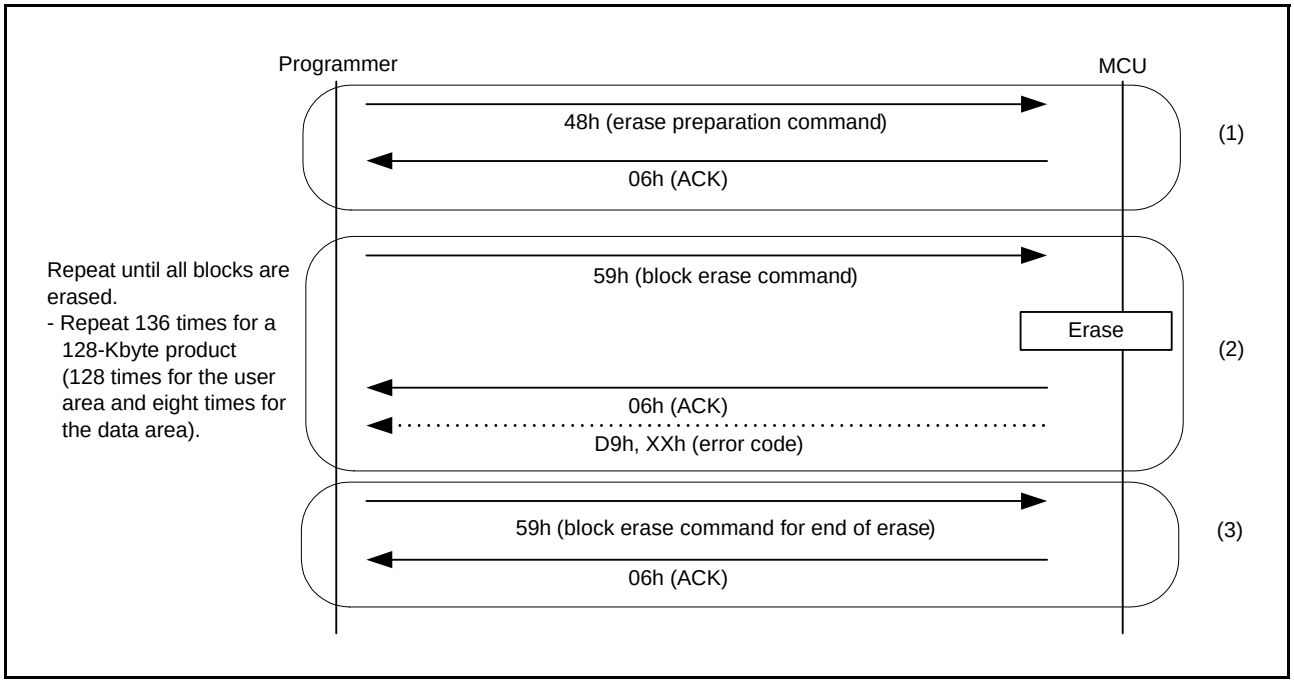



**Figure 35.19 Procedure to Send Commands in Erase Ready Operation**

### 35.9.10.7 Erase the User Area and Data Area

Erase blocks that are programmed in the user area and data area to program a user program.

(1) Send an erase preparation command (48h) to place the MCU in the erase wait state. The MCU enters the erase wait state and sends a response (06h).

(2) Send a block erase command (59h). Set the block that is erased in the block start address. When the selected block is erased successfully, the MCU sends a response (06h). When the MCU fails to receive, the MCU sends an error response (D9h).

(3) In order to place the MCU in the program/erase state, send a block erase command for end of erase (59h 04h FFh FFh FFh FFh A7h). The MCU enters the program/erase state and sends a response (06h).

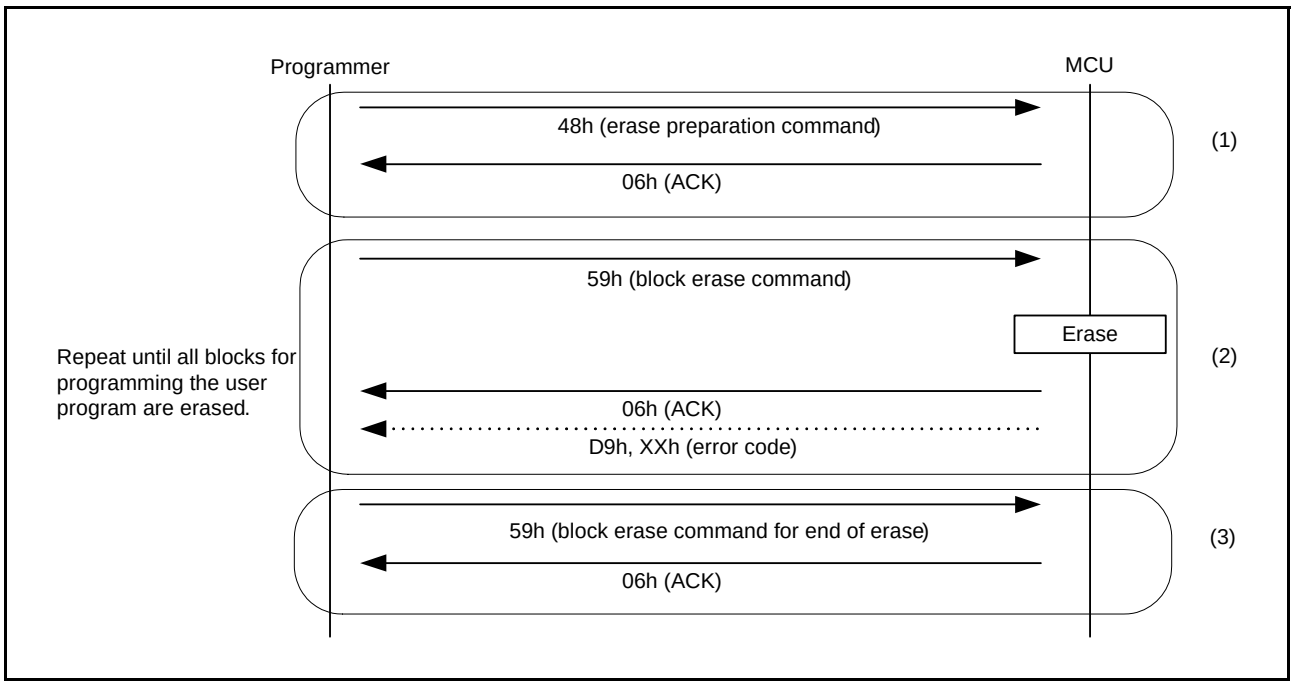



**Figure 35.20 Procedure to Erase the User Area and Data Area**

### 35.9.10.8 Program the User Area and Data Area

Program a user program in the user area and data area.

(1) Send the user/data area program preparation command (43h) to place the MCU in the program wait state. The MCU enters the program wait sate and sends a response (06h).

(2) Send the program command (50h). Set the program address to an address aligned on a 256-byte boundary. Set program data in 256 bytes. When the data is programmed successfully, the MCU sends a response (06h). When the MCU fails to receive, the MCU sends an error response (D0h).

(3) In order to place the MCU in the program/erase state, send the program command for end of program (50h FFh FFh FFh FFh B4h). The MCU enters the program/erase state and sends a response (06h).

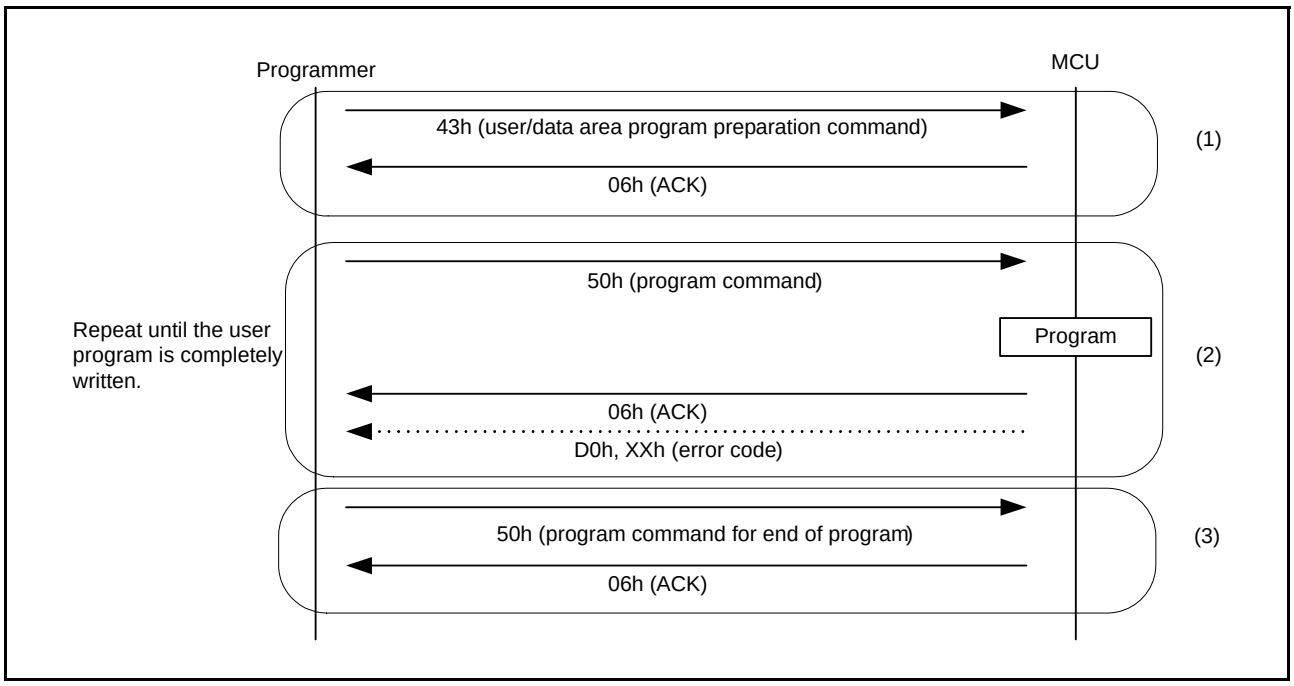



**Figure 35.21 Procedure to Program the User Area and Data Area**

### 35.9.10.9 Check Data in the User Area

Read and check, checksum, and blank check the user area to check the programmed data in the user area.

(1) The read and check operation is used to read data in the user area and compare the read data with the programmed data to check if the program operation is performed successfully. Send a memory read command (52h) to read data in the user area. Set the read area in the read address and read size. The MCU sends data for the size from the address set in the read address. If the MCU fails to receive, the MCU sends an error response (D2h).

(2) Send the user area checksum command (4Bh) to check program data using the checksum of user area. The MCU reads data from the start address to the end address of the user area in bytes and send the read result as a response.

(3) Send a user area blank check command (4Dh) to check if the user area has data. When there is no data in the start address to the end address of the user area, the MCU sends a response (06h). When there is at least 1 byte of data, the MCU sends a response (CDh, 52h) indicating that the selected area is not blank.

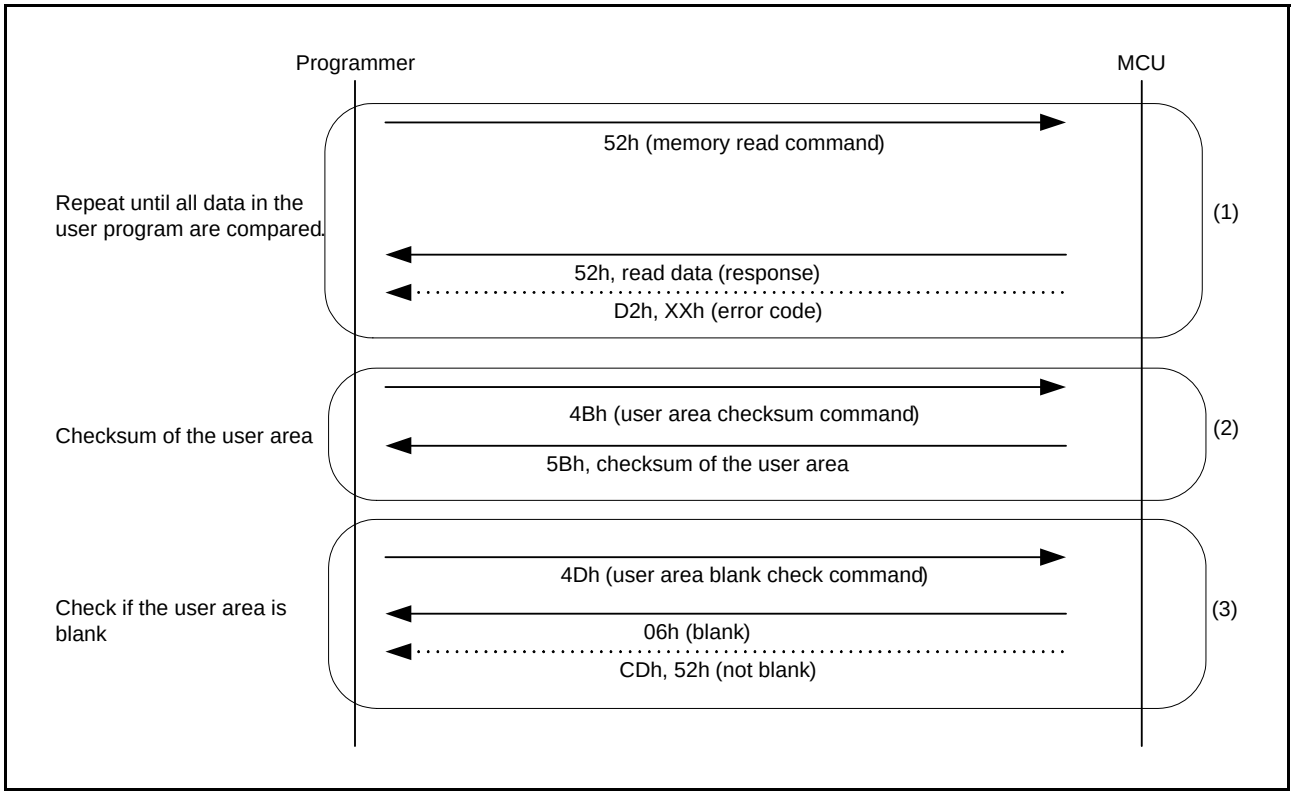



**Figure 35.22 Procedure to Check Data in the User Area**

### 35.9.10.10 Check Data in the Data Area

Read and check, checksum, and blank check the user area to check the programmed data in the data area.

(1) The read and check operation is used to read data in the data area and compare the read data with the programmed data to check if the program operation is performed successfully. Send a memory read command (52h) to read data in the data area. Set the read area in the read address and read size. The MCU sends data for the size from the address set in the read address. If the MCU fails to receive, the MCU sends an error response (D2h).

(2) Send the data area checksum command (61h) to check program data using the checksum of data area. The MCU reads data from the start address to the end address of the data area in bytes and send the read result as a response.

(3) Send the data area blank check command (62h) to check if the data area has data. When there is no data in the start address to the end address of the data area, the MCU sends a response (06h). When there is at least 1 byte of data, the MCU sends a response (E2h, 52h) indicating that the selected area is not blank.

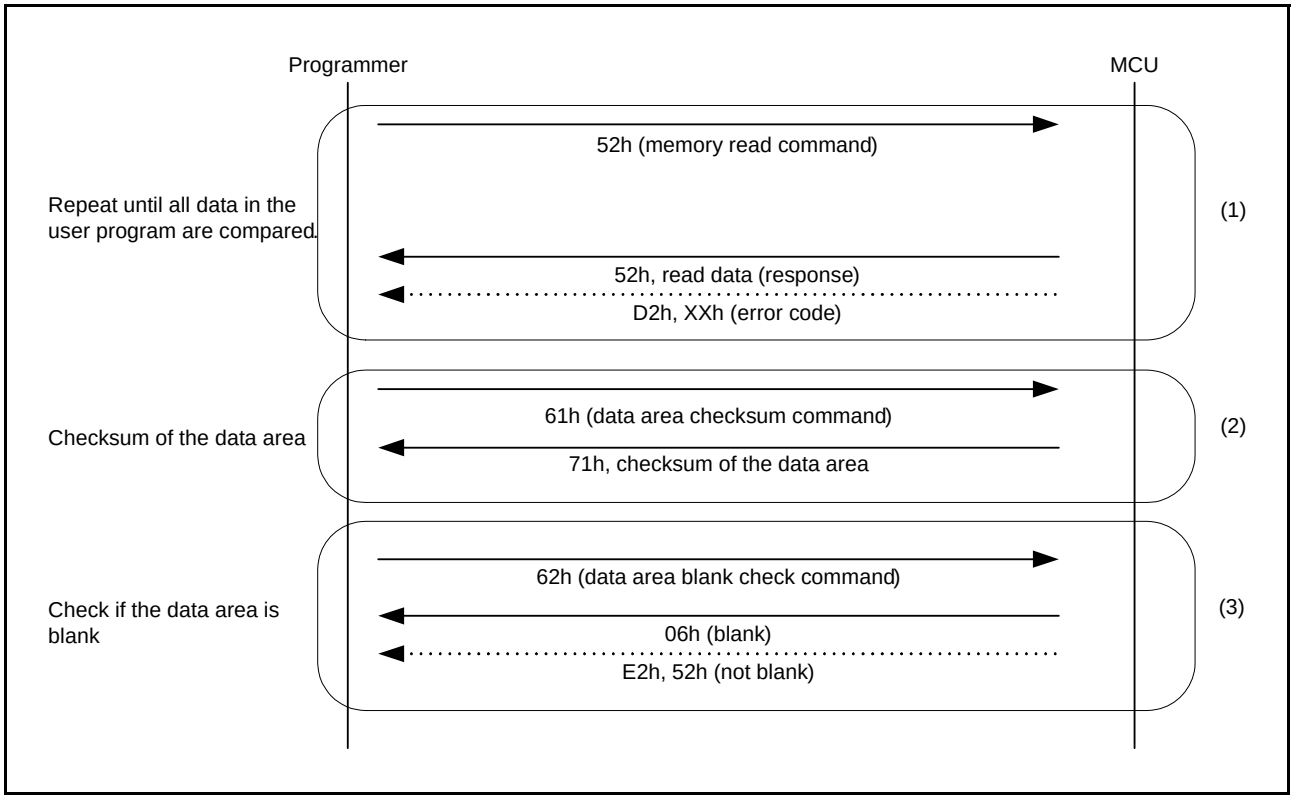



**Figure 35.23 Procedure to Check Data in the Data Area**

### 35.9.10.11 Set the Access Window in the User Area

Set the access window to avoid unintentionally rewriting the user area by the self-programming library.

(1) When setting the access window, set 00h in the access window, set the start address of the area that can be programmed the by self-programming library in the access window start address LH and access window start address HL, and set the end address of the area that can be programmed by the self-programming library in the access window end address LH and access window end address HL.
When clearing the access window settings, set FFh in the access window, access window start address LH, access window start address HL, the access window end address LH, and the access window end address HL.
When the MCU writes the addresses, the MCU sends a response (06h). If the MCU fails to receive, the MCU sends an error response (F4h).

(2) Send the access window read command (73h) to confirm the access window settings. The MCU sends the current access window settings. When the MCU fails to receive, the MCU sends an error response (F3h).

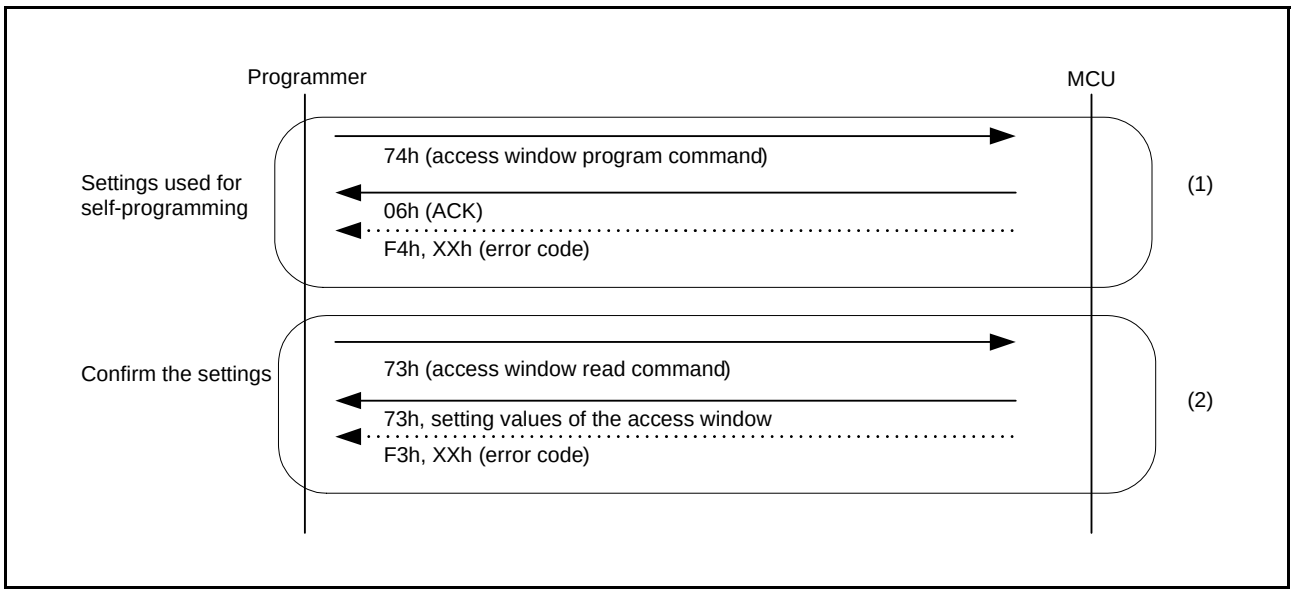



**Figure 35.24 Procedure to Set the Access Window in the User Area**

## 35.10 Rewriting by Self-Programming

### 35.10.1 Overview

The MCU supports rewriting the flash memory by the user program. Using the self-programming library provided from Renesas Electronics, the ROM and E2 DataFlash can be rewritten.
When rewriting the E2 DataFlash, the BGO can be used to execute the rewrite program on the ROM. The E2 DataFlash can also be rewritten by executing the rewrite program that is transferred on the RAM in advance.

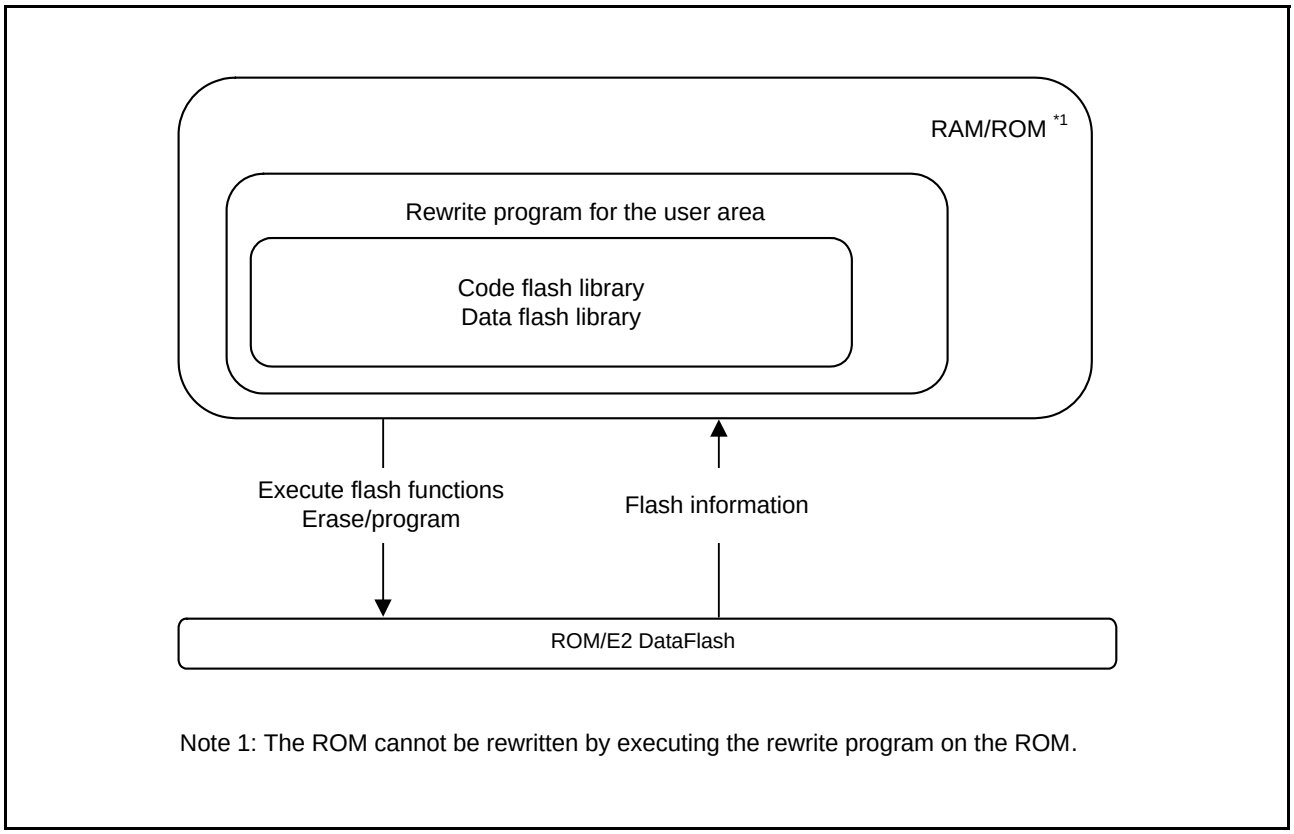



**Figure 35.25 Self-Programming Overview**

Refer to the user's manual of "Code Flash Libraries" and "Data Flash Libraries" for comprehensive information about flash self-programming.

## 35.11 Usage Notes

(1) Erase Suspended Area
Data in areas where an erase operation is suspended is undefined. To avoid malfunctions caused by reading undefined data, do not execute commands and read data in the area where an erase operation is suspended.

(2) Suspension by Erase Suspend Commands
When suspending an erase operation by the erase suspend command, complete the operation by a resume command.

(3) Additional Programming Disabled
The same address cannot be programmed more than once. When programming an area that has been already programmed, erase the area first.

(4) Reset during Program/Erase
If inputting a reset from the RES# pin, release the reset after reset input time of at least tRESW (refer to section 36, Electrical Characteristics) within the range of the operating voltage defined in the electrical characteristics.
The IWDT reset and software reset can be used regardless of tRESW.

(5) Non-maskable Interrupt Disabled during Program/Erase
When a non-maskable interrupt (NMI pin interrupt, oscillation stop detection interrupt, IWDT underflow/refresh error, voltage monitoring 1 interrupt, or voltage monitoring 2 interrupt) occurs during a program/erase operation, the vectors are fetched from the ROM, and undefined data is read. Therefore, do not generate a non-maskable interrupt during a program/erase operation on the ROM.
(The description in (5) applies only to the ROM.)

(6) Location of Interrupt Vectors during a Program/Erase Operation
When an interrupt occurs during a program/erase operation, the vector may be fetched from the ROM. To avoid fetching the vector from the ROM, set the destination for fetching interrupt vectors to an area other than the ROM with the CPU interrupt table register (INTB).

(7) Program/Erase in Low-Speed Operating Mode
Do not program or erase the flash memory when low-speed operating mode is selected with the sub operating power control register (SOPCCR).

(8) Abnormal Termination during Program/Erase
When the voltage exceeds the range of the operating voltage during a program/erase operation or when a program/erase operation is not completed successfully due to a reset or prohibited actions described in (9), erase the area again.

(9) Actions Prohibited during Program/Erase
To prevent the damage to the flash memory, comply with the following instructions.

- Do not use the MCU power supply that is outside the operating voltage range.
- Do not update the value of the OPCCR.OPCM[2:0] bits.
- Do not update the value of the SOPCCR.SOPCM bit.
- Do not change the clock source select bit in the SCKCR3 register.
- Do not enable switching clock sources by setting the RSTCKCR.RSTCKEN bit when exiting sleep mode.
- Do not change the division ratio of the flash interface clock (FCLK).
- Do not place the MCU in deep sleep mode or software standby mode.
- Do not access the E2 DataFlash during a program/erase operation to the ROM.
- Do not change the DFLCTL.DFLEN bit value during a program/erase operation to the E2 DataFlash.

(10) FCLK during Program/Erase
When using the self-programming library for program/erase, the FCLK operating frequency needs to be set. Set the frequency to the integer value. If a frequency other than the selected frequency is used, program time or time to apply stress to memory cells may be longer and therefore the program time or the reliability of Flash Macro may be affected. Note that when the FCLK is 4 to 32 MHz, a rounded-up value should be set for a non-integer frequency such as 12.5 MHz (i.e. 12.5 MHz should be set rounded up to 13 MHz).

## 35.12 Usage Notes in Boot Mode

(1) Notes on Communication Errors in Boot Mode
When communication with the MCU cannot be performed properly, reset and start up in boot mode again.

(2) Notes on Power Supply Voltage in SCI Mode
When the bit rate exceeds 500 kbps in SCI mode, use a voltage that is 3.0 or higher, and 3.6 V or lower.

(3) Notes on Using SCI Mode and USB0 Self-Powered Mode
When SCI mode is selected, do not connect a USB cable.

(4) Notes on Using USB0 Battery Charging Function
Do not select SCI mode.

(5) Notes on Option-Setting Memory in Boot Mode
The settings of option function select register 0 (OFS0), option function select register 1 (OFS1), and endian select register (MDE) are invalid in boot mode.

(6) Notes on Clocks in USB Interface Mode
When USB interface mode is selected, externally input a clock to the EXTAL or XTAL pin, or connect a crystal or ceramic resonator to supply a clock.
Use a 6 MHz, 8 MHz, 12 MHz, or 16 MHz external clock in USB interface mode. An clock other than a 6 MHz, 8 MHz, 12 MHz, or 16 MHz external clock cannot be used.

(7) Notes on Power Supply Voltage in USB Interface Mode
Use a voltage that is 3.0 V or higher, and 3.6 V or lower in USB interface mode. A voltage that is 1.8 V or higher, and lower than 3.0 V cannot be used.

# 36. Electrical Characteristics

## 36.1 Absolute Maximum Ratings

**Table 36.1 Absolute Maximum Ratings**

Conditions: VSS = AVSS0 = VREFL0 = VSS_USB = 0 V

| Item | Symbol | Value | Unit |
|---|---|---|---|
| Power supply voltage | VCC, VCC_USB | –0.3 to +4.6 | V |
| Input voltage (except for ports for 5 V tolerant*1) | $V_{in}$ | –0.3 to VCC +0.3 | V |
| Input voltage (ports for 5 V tolerant*1) | $V_{in}$ | –0.3 to +6.5 | V |
| Reference power supply voltage | VREFH0 | –0.3 to AVCC +0.3 | V |
| Analog power supply voltage | AVCC0 | –0.3 to +4.6 | V |
| Analog input voltage | $V_{AN}$ | –0.3 to +4.6 | V |
| Operating temperature*2 | $T_{opr}$ | –0.3 to AVCC + 0.3<br>(when AN000 to AN004 and AN006 used)<br>–0.3 to VCC + 0.3<br>(when AN008 to AN015 used) | °C |
| Storage temperature | $T_{stg}$ | –55 to +125 | °C |

Caution: Permanent damage to the MCU may result if absolute maximum ratings are exceeded.

To preclude any malfunctions due to noise interference, insert capacitors of high frequency characteristics between the VCC and VSS pins, between the AVCC0 and AVSS0 pins, between the VCC_USB and VSS_USB pins, and between the VREFH0 and VREFL0 pins. Place capacitors of about 0.1 µF as close as possible to every power supply pin and use the shortest and heaviest possible traces. Also, connect capacitors as stabilization capacitance.

Connect the VCL pin to a VSS pin via a 4.7 µF capacitor. The capacitor must be placed close to the pin.

Note 1. Ports 16, 17, A6, and B0 are 5 V tolerant.
Do not input signals or an I/O pull-up power supply while the device is not powered. The current injection that results from input of such a signal or I/O pull-up may cause malfunction and the abnormal current that passes in the device at this time may cause degradation of internal elements.

Note 2. The upper limit of operating temperature is 85°C or 105°C, depending on the product. For details, refer to 1.2 List of Products.

**Table 36.2 Recommended Operating Voltage Conditions**

| Item | Symbol | Value | Unit |
|---|---|---|---|
| Recommended operating voltage conditions | VCC, VCC_USB*1 | 1.8 to 3.6 (during no USB communication)<br>3.0 to 3.6 (during USB communication) | V |
| | AVCC0*2 | 1.8 to 3.6 | V |

Note 1. Set VCC and VCC_USB to the same potential. Also, set VSS, AVSS0, and VSS_USB to the same potential.

Note 2. AVCC0 and VCC can be set individually within the operating range. For details, 30.7.10 Voltage Range of Analog Power Supply Pins.

## 36.2 DC Characteristics

**Table 36.3 DC Characteristics (1)**

Conditions: VCC = AVCC0 = VCC_USB = 2.7 to 3.6 V, VSS = AVSS0 = VREFL0 = VSS_USB = 0 V, $T_a$ = –40 to +105°C

| Item | | Symbol | Min. | Typ. | Max. | Unit | Test Conditions |
|---|---|---|---|---|---|---|---|
| Schmitt trigger input voltage | RIIC input pin (except for SMBus, 5 V tolerant) | $V_{IH}$ | VCC × 0.7 | — | 5.8 | V | |
| | Ports 16, 17, port A6, port B0 (5 V tolerant) | | VCC × 0.8 | — | 5.8 | | |
| | Other pins Ports 03, 05, ports 14,15 ports 26, 27, ports 30 to 32, 35 ports 54, 55, ports A0, A1, A3, A4 ports B1, B3, B5 to B7 ports C2 to C7 port E, ports H6, H7, RES# | | VCC × 0.8 | — | VCC + 0.3 | | |
| | RIIC input pin (except for SMBus) | $V_{IL}$ | –0.3 | — | VCC × 0.3 | | |
| | Other than RIIC input pin | | –0.3 | — | VCC × 0.2 | | |
| | RIIC input pin (except for SMBus) | $\Delta V_T$ | VCC × 0.05 | — | — | | |
| | Other than RIIC input pin | | VCC × 0.1 | — | — | | |
| Input level voltage (except for Schmitt trigger input pins) | MD | $V_{IH}$ | VCC × 0.9 | — | VCC + 0.3 | V | |
| | XTAL (external clock input) | | VCC × 0.8 | — | VCC + 0.3 | | |
| | Ports 40 to 44, 46, ports J6, J7 | | VCC × 0.7 | — | VCC + 0.3 | | |
| | RIIC input pin (SMBus) | | 2.1 | — | VCC + 0.3 | | |
| | MD | $V_{IL}$ | –0.3 | — | VCC × 0.1 | | |
| | XTAL (external clock input) | | –0.3 | — | VCC × 0.2 | | |
| | Ports 40 to 44, 46, ports J6, J7 | | –0.3 | — | VCC × 0.3 | | |
| | RIIC input pin (SMBus) | | –0.3 | — | 0.8 | | |

**Table 36.4 DC Characteristics (2)**

Conditions: VCC = AVCC0 = VCC_USB = 1.8 to 2.7 V, VSS = AVSS0 = VREFL = VREFL0 = VSS_USB = 0 V, $T_a$ = –40 to +105°C

| Item | | Symbol | Min. | Typ. | Max. | Unit | Test Conditions |
|---|---|---|---|---|---|---|---|
| Schmitt trigger input voltage | Ports 16, 17, port A6, port B0 (5 V tolerant) | $V_{IH}$ | VCC × 0.8 | — | 5.8 | V | |
| | Other pins Ports 03, 05, ports 14,15 ports 26, 27, ports 30 to 32, 35 ports 54, 55, ports A0, A1, A3, A4 ports B1, B3, B5 to B7 ports C2 to C7 port E, ports H6, H7, RES# | | VCC × 0.8 | — | VCC + 0.3 | | |
| | All pins | | –0.3 | — | VCC × 0.2 | | |
| | All pins | $\Delta V_T$ | VCC × 0.01 | — | — | | |
| Input level voltage (except for Schmitt trigger input pins) | MD | $V_{IH}$ | VCC × 0.9 | — | VCC + 0.3 | V | |
| | XTAL (external clock input) | | VCC × 0.8 | — | VCC + 0.3 | | |
| | Ports 40 to 44, 46, ports J6, J7 | | VCC × 0.7 | — | VCC + 0.3 | | |
| | MD | $V_{IL}$ | –0.3 | — | VCC × 0.1 | | |
| | XTAL (external clock input) | | –0.3 | — | VCC × 0.2 | | |
| | Ports 40 to 44, 46, ports J6, J7 | | –0.3 | — | VCC × 0.3 | | |

**Table 36.5 DC Characteristics (3)**

Conditions: VCC = AVCC0 = VCC_USB = 1.8 to 3.6 V, VSS = AVSS0 = VREFL0 = VSS_USB = 0 V, $T_a$ = –40 to +105°C

| Item | | Symbol | Min. | Typ. | Max. | Unit | Test Conditions |
|---|---|---|---|---|---|---|---|
| Input leakage current | RES#, MD, port 35 ports H6, H7 | $\lvert I_{in} \rvert$ | — | — | 1.0 | µA | $V_{in}$ = 0 V, VCC |
| Three-state leakage current (off-state) | Ports for 5 V tolerant | $\lvert I_{TSI} \rvert$ | — | — | 1.0 | µA | $V_{in}$ = 0 V, 5.8 V |
| | Pins other than above | | — | — | 1.0 | | $V_{in}$ = 0 V, VCC |
| Input capacitance | All input pins (except for port 35, port 16, USB0_DM, USB0_DP) | $C_{in}$ | — | — | 15 | pF | $V_{in}$ = 0 mV, f = 1 MHz, $T_a$ = 25°C |
| | Port 35, port 16, USB0_DM, USB0_DP | | — | — | 30 | | |

**Table 36.6 DC Characteristics (4)**

Conditions: VCC = AVCC0 = VCC_USB = 1.8 to 3.6 V, VSS = AVSS0 = VREFL0 = VSS_USB = 0 V, $T_a$ = –40 to +105°C

| Item | | Symbol | Min. | Typ. | Max. | Unit | Test Conditions |
|---|---|---|---|---|---|---|---|
| Input pull-up resistor | All ports (except for ports 35, PH7) | $R_U$ | 10 | 20 | 100 | kΩ | $V_{in}$ = 0 V |

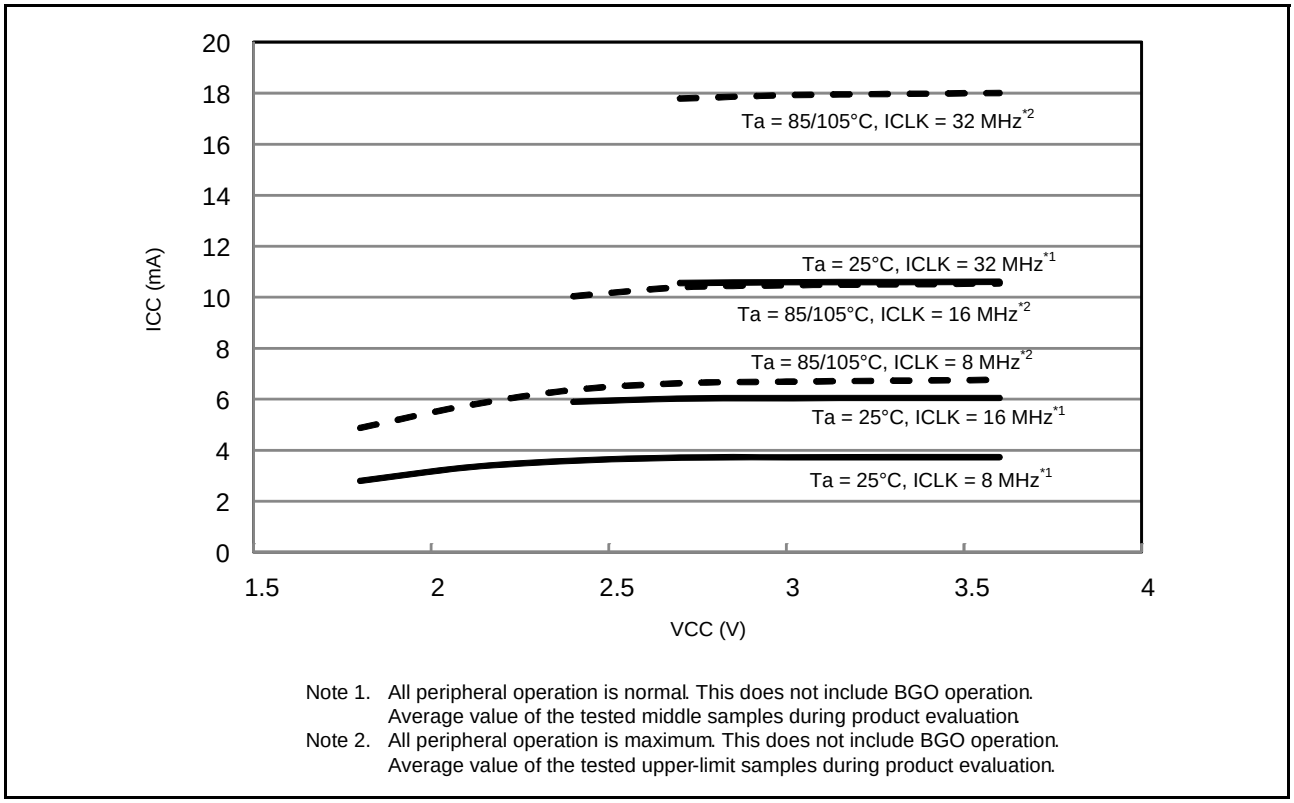
**Table 36.7 DC Characteristics (5)**

Conditions: VCC = AVCC0 = VCC_USB = 1.8 to 3.6 V, VSS = AVSS0 = VREFL0 = VSS_USB = 0 V, $T_a$ = –40 to +105°C

| Item | | | | | Symbol | Typ *4 | Max | Unit | Test Conditions |
|---|---|---|---|---|---|---|---|---|---|
| Supply current*1 | High-speed operating mode | Normal operating mode | No peripheral operation*2 | ICLK = 32 MHz | $I_{CC}$ | 3.2 | — | mA | |
| | | | | ICLK = 16 MHz | | 2.2 | — | | |
| | | | | ICLK = 8 MHz | | 1.7 | — | | |
| | | | All peripheral operation: Normal*3 | ICLK = 32 MHz | | 10.6 | — | | |
| | | | | ICLK = 16 MHz | | 6.1 | — | | |
| | | | | ICLK = 8 MHz | | 3.7 | — | | |
| | | | All peripheral operation: Max.*3 | ICLK = 32 MHz | | — | 24 | | |
| | | Sleep mode | No peripheral operation*2 | ICLK = 32 MHz | | 1.8 | — | | |
| | | | | ICLK = 16 MHz | | 1.4 | — | | |
| | | | | ICLK = 8 MHz | | 1.1 | — | | |
| | | | All peripheral operation: Normal*3 | ICLK = 32 MHz | | 6.4 | — | | |
| | | | | ICLK = 16 MHz | | 3.7 | — | | |
| | | | | ICLK = 8 MHz | | 2.4 | — | | |
| | | Deep sleep mode | No peripheral operation*2 | ICLK = 32 MHz | | 1.2 | — | | |
| | | | | ICLK = 16 MHz | | 1.0 | — | | |
| | | | | ICLK = 8 MHz | | 0.90 | — | | |
| | | | All peripheral operation: Normal*3 | ICLK = 32 MHz | | 4.6 | — | | |
| | | | | ICLK = 16 MHz | | 2.8 | — | | |
| | | | | ICLK = 8 MHz | | 1.8 | — | | |
| | | Increase during BGO operation*5 | | | | 2.5 | — | | |
| | Middle-speed operating modes | Normal operating mode | No peripheral operation*6 | ICLK = 12 MHz | $I_{CC}$ | 2.0 | — | mA | |
| | | | | ICLK = 8 MHz | | 1.3 | — | | |
| | | | | ICLK = 1 MHz | | 0.75 | — | | |
| | | | All peripheral operation: Normal*7 | ICLK = 12 MHz | | 4.9 | — | | |
| | | | | ICLK = 8 MHz | | 3.5 | — | | |
| | | | | ICLK = 1 MHz | | 1.2 | — | | |
| | | | All peripheral operation: Max.*7 | ICLK = 12 MHz | | — | 11 | | |
| | | Sleep mode | No peripheral operation*6 | ICLK = 12 MHz | | 1.4 | — | | |
| | | | | ICLK = 8 MHz | | 0.85 | — | | |
| | | | | ICLK = 1 MHz | | 0.65 | — | | |
| | | | All peripheral operation: Normal*7 | ICLK = 12 MHz | | 3.2 | — | | |
| | | | | ICLK = 8 MHz | | 2.2 | — | | |
| | | | | ICLK = 1 MHz | | 1.0 | — | | |
| | | Deep sleep mode | No peripheral operation*6 | ICLK = 12 MHz | | 1.2 | — | | |
| | | | | ICLK = 8 MHz | | 0.70 | — | | |
| | | | | ICLK = 1 MHz | | 0.60 | — | | |
| | | | All peripheral operation: Normal*7 | ICLK = 12 MHz | | 2.5 | — | | |
| | | | | ICLK = 8 MHz | | 1.8 | — | | |
| | | | | ICLK = 1 MHz | | 0.90 | — | | |
| | | Increase during BGO operation*5 | | | | 2.5 | — | | |

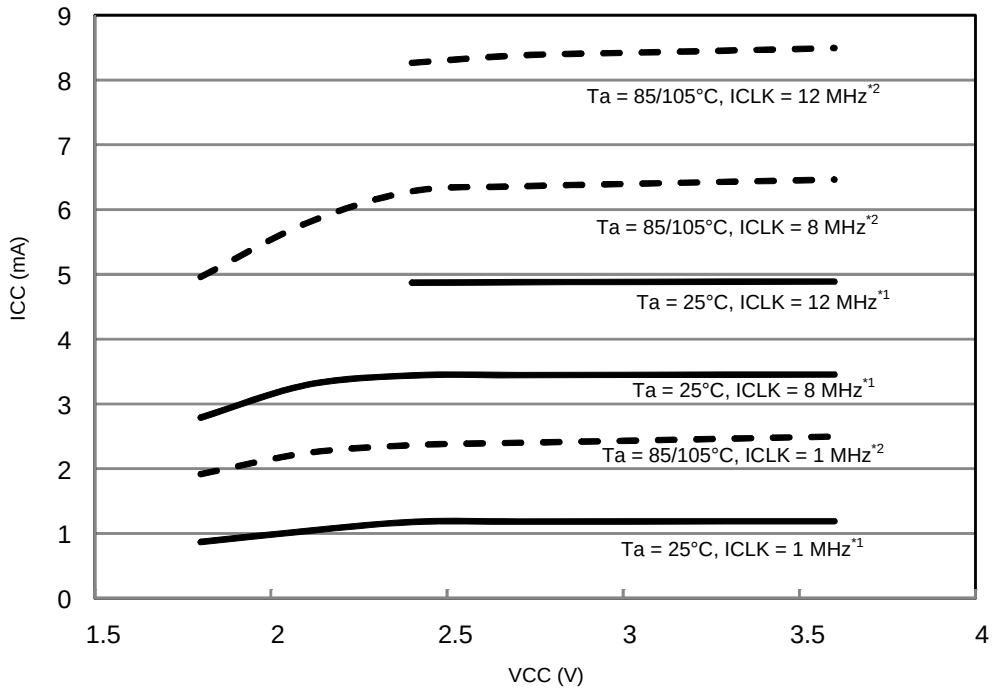
| Item | | | | | Symbol | Typ *4 | Max | Unit | Test Conditions |
|---|---|---|---|---|---|---|---|---|---|
| Supply current*1 | Low-speed operating mode | Normal operating mode | No peripheral operation*8 | ICLK = 32.768 kHz | $I_{CC}$ | 4.0 | — | μA | |
| | | | All peripheral operation: Normal*9, *10 | ICLK = 32.768 kHz | | 11.5 | — | | |
| | | | All peripheral operation: Max.*9, *10 | ICLK = 32.768 kHz | | — | 40 | | |
| | | Sleep mode | No peripheral operation*8 | ICLK = 32.768 kHz | | 2.2 | — | | |
| | | | All peripheral operation: Normal*9 | ICLK = 32.768 kHz | | 7.1 | — | | |
| | | Deep sleep mode | No peripheral operation*8 | ICLK = 32.768 kHz | | 1.8 | — | | |
| | | | All peripheral operation: Normal*9 | ICLK = 32.768 kHz | | 5.3 | | | |

Note 1. Supply current values do not include output charge/discharge current from all pins. The values apply when internal pull-up MOSs are in the off state.
Note 2. Clock supply to the peripheral functions is stopped. This does not include BGO operation. The clock source is PLL. FCLK and PCLK are set to divided by 64.
Note 3. Clocks are supplied to the peripheral functions. This does not include BGO operation. The clock source is PLL. FCLK and PCLK are set to the same frequency as ICLK.
Note 4. Values when VCC = 3.3 V.
Note 5. This is the increase for programming or erasure of the ROM or flash memory for data storage during program execution.
Note 6. Clock supply to the peripheral functions is stopped. The clock source is PLL when ICLK = 12 MHz, and HOCO otherwise. FCLK and PCLK are set to divided by 64.
Note 7. Clocks are supplied to the peripheral functions. The clock source is PLL when ICLK = 12 MHz, and HOCO otherwise. FCLK and PCLK are set to the same frequency as ICLK.
Note 8. Clock supply to the peripheral functions is stopped. The clock source is the sub-clock oscillator. FCLK and PCLK are set to divided by 64.
Note 9. Clocks are supplied to the peripheral functions. The clock source is the sub-clock oscillator. FCLK and PCLK are set to the same frequency as ICLK.
Note 10. Values when the MSTPCRA.MSTPA17 bit (12-bit A/D converter module stop bit) is set to “transition to the module stop state is made”.



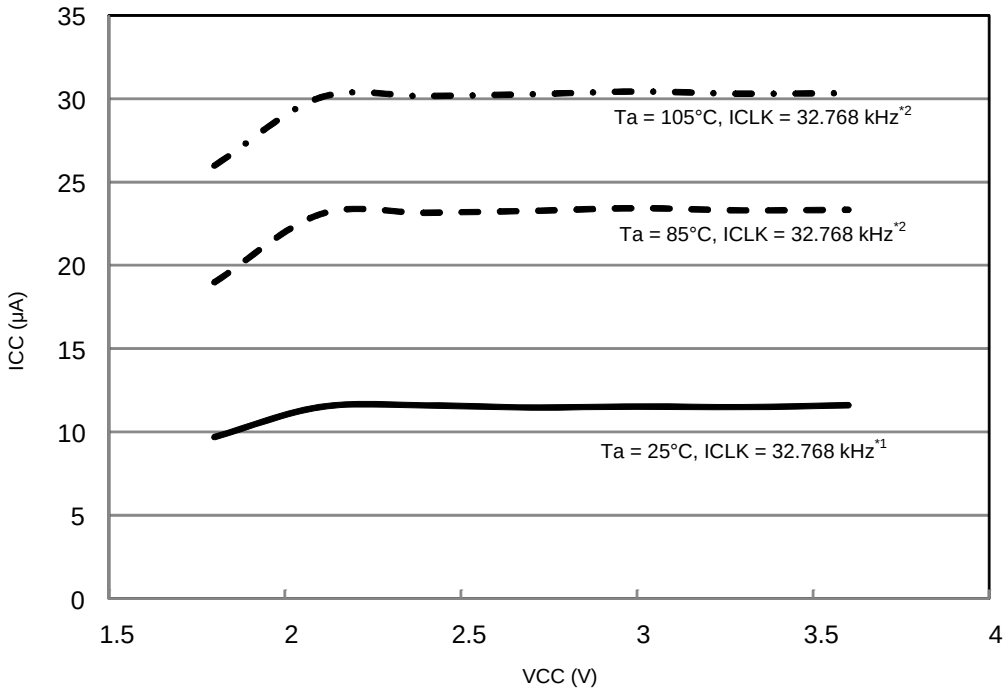

Note 1. All peripheral operation is normal. This does not include BGO operation.
Average value of the tested middle samples during product evaluation.
Note 2. All peripheral operation is maximum. This does not include BGO operation.
Average value of the tested upper-limit samples during product evaluation.

**Figure 36.1 Voltage Dependency in High-Speed Operating Mode (Reference Data)**




Note 1. All peripheral operation is normal. This does not include BGO operation.
Average value of the tested middle samples during product evaluation.
Note 2. All peripheral operation is maximum. This does not include BGO operation.
Average value of the tested upper-limit samples during product evaluation.

**Figure 36.2 Voltage Dependency in Middle-Speed Operating Mode (Reference Data)**




Note 1. All peripheral operation is normal.
Average value of the tested middle samples during product evaluation.
Note 2. All peripheral operation is maximum.
Average value of the tested upper-limit samples during product evaluation.

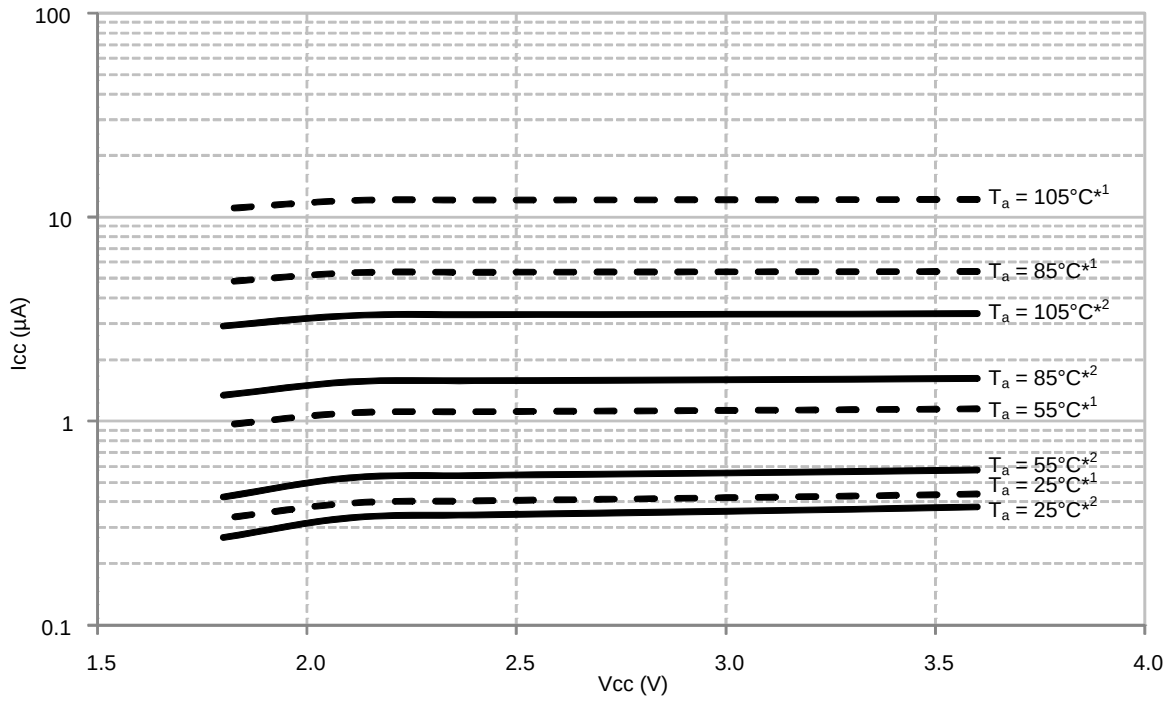
**Figure 36.3 Voltage Dependency in Low-Speed Operating Mode (Reference Data)**

**Table 36.8 DC Characteristics (6)**

Conditions: VCC = AVCC0 = VCC_USB = 1.8 to 3.6 V, VSS = AVSS0 = VSS_USB = 0 V, $T_a$ = –40 to +105°C

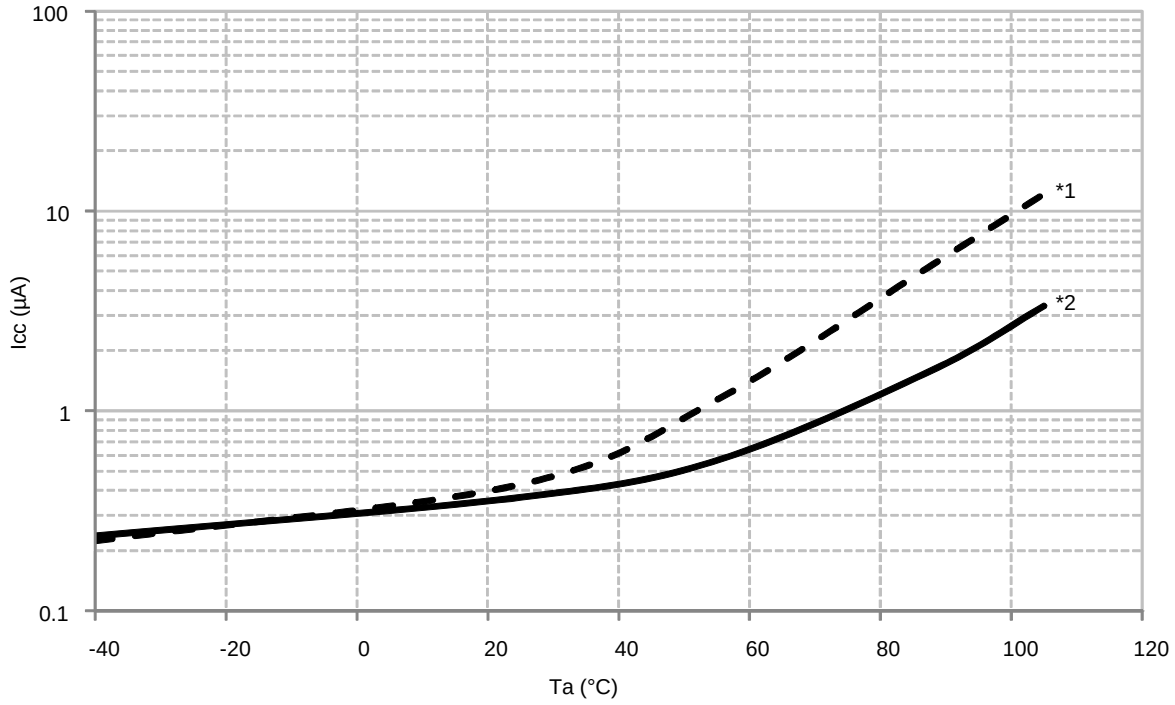
| Item | | | Symbol | Typ.*3 | Max. | Unit | Test Conditions |
|---|---|---|---|---|---|---|---|
| Supply current*1 | Software standby mode*2 | $T_a$ = 25°C | $I_{CC}$ | 0.35 | 0.53 | µA | |
| | | $T_a$ = 55°C | | 0.58 | 1.45 | | |
| | | $T_a$ = 85°C | | 1.60 | 7.30 | | |
| | | $T_a$ = 105°C | | 3.30 | 16.50 | | |
| | Increment for RTC operation*4 | | | 0.31 | — | | RCR3.RTCDV[1:0] set to low drive capacity |

Note 1. Supply current values are with all output pins unloaded and all input pull-up MOSs in the off state.
Note 2. The IWDT and LVD are stopped.
Note 3. VCC = 3.3 V.
Note 4. Includes the oscillation circuit.




Note1. Average value of the tested upper-limit samples during product evaluation.
Note2. Average value of the tested middle samples during product evaluation.

**Figure 36.4 Voltage Dependency in Software Standby Mode (Reference Data)**




Note1. Average value of the tested upper-limit samples during product evaluation.
Note2. Average value of the tested middle samples during product evaluation.

**Figure 36.5 Temperature Dependency in Software Standby Mode (Reference Data)**

### Table 36.9 DC Characteristics (7)

Conditions: Products with operating temperature ($T_a$) –40 to +105°C
VCC = AVCC0 = VCC_USB = 1.8 to 3.6 V, VSS = AVSS0 = VREFL0 = VSS_USB = 0 V

| Item | Symbol | Typ. | Max. | Unit | Test Conditions |
|---|---|---|---|---|---|
| Permissible junction temperature | Tj | — | 120 | °C | High-speed operating mode |
| | | — | 105 | | Middle-speed operating mode |
| | | — | 120 | | Low-speed operating mode |

Note: • Make sure that Tj < Ta + 0.1 * total power consumption (mW), where total power consumption = (VCC - $V_{OH}$) × $\Sigma I_{OH}$ + $V_{OL}$ × $\Sigma I_{OL}$ + $I_{CC}$max × VCC.

### Table 36.10 DC Characteristics (8)

Conditions: Products with operating temperature ($T_a$) –40 to +85°C
VCC = AVCC0 = VCC_USB = 1.8 to 3.6 V, VSS = AVSS0 = VREFL0 = VSS_USB = 0 V

| Item | Symbol | Typ. | Max. | Unit | Test Conditions |
|---|---|---|---|---|---|
| Permissible junction temperature | Tj | — | 120 | °C | High-speed operating mode |
| | | — | 105 | | Middle-speed operating mode |
| | | — | 120 | | Low-speed operating mode |

Note: • Make sure that Tj < Ta + 0.1 * total power consumption (mW), where total power consumption = (VCC - $V_{OH}$) × $\Sigma I_{OH}$ + $V_{OL}$ × $\Sigma I_{OL}$ + $I_{CC}$max × VCC.

### Table 36.11 DC Characteristics (9)

Conditions: VCC = AVCC0 = VCC_USB = 1.8 to 3.6 V, VSS = AVSS0 = VREFL0 = VSS_USB = 0 V, $T_a$ = –40 to +105°C

| Item | | Symbol | Min. | Typ. | Max. | Unit | Test Conditions |
|---|---|---|---|---|---|---|---|
| Analog power supply current | During A/D conversion | $I_{AVCC}$ | — | 0.7 | 1.2 | mA | |
| | During D/A conversion (per channel)*1 | | — | — | 1.5 | | |
| | Temperature sensor | | — | 75 | — | µA | |
| | Waiting for A/D, D/A conversion (all units) | | — | — | 0.3 | | |
| Reference power supply current | During A/D conversion | $I_{REFH0}$ | — | 25 | 52 | µA | |
| | Waiting for A/D conversion (all units) | | — | — | 60 | nA | |
| USB operating current | During USB communication operation under the following settings and conditions<br>• Host controller operation is set to full-speed mode<br>Bulk OUT transfer (64 bytes) × 1,<br>bulk IN transfer (64 bytes) × 1<br>• Connect peripheral devices via a 1-meter USB cable from the USB port. | $I_{USBH}$*2 | — | 4.3 (VCC)<br>0.9 (USB_VCC) | — | mA | |
| | During USB communication operation under the following settings and conditions<br>• Function controller operation is set to full-speed mode<br>Bulk OUT transfer (64 bytes) × 1,<br>bulk IN transfer (64 bytes) × 1<br>• Connect the host device via a 1-meter USB cable from the USB port. | $I_{USBF}$*2 | — | 3.6 (VCC)<br>1.1 (USB_VCC) | — | mA | |
| | During suspended state under the following setting and conditions<br>• Function controller operation is set to full-speed mode (pull up the UDP0 pin)<br>• Software standby mode<br>• Connect the host device via a 1-meter USB cable from the USB port. | $I_{SUSP}$*3 | — | 0.35 (VCC)<br>170 (USB_VCC) | — | µA | |

Note 1. The reference power supply current is included in the power supply current value for D/A conversion.
Note 2. Current consumed only by the USB module.
Note 3. Includes the current supplied from the pull-up resistor of the USB0_DP pin to the pull-down resistor of the host device, in addition to the current consumed by this MCU during the suspended state.

### Table 36.12 DC Characteristics (10)

Conditions: VCC = AVCC0 = VCC_USB = 1.8 to 3.6 V, VSS = AVSS0 = VREFL0 = VSS_USB = 0 V, $T_a$ = –40 to +105°C

| Item | Symbol | Min. | Typ. | Max. | Unit | Test Conditions |
|---|---|---|---|---|---|---|
| RAM standby voltage | $V_{RAM}$ | 1.8 | — | 3.6 | V | |

### Table 36.13 DC Characteristics (11)

Conditions: VCC = AVCC0 = VCC_USB = 0 to 3.6 V, VREFH0 = 0 to AVCC0, VSS = AVSS0 = VREFL0 = VSS_USB = 0 V, $T_a$ = –40 to +105°C

| Item | | Symbol | Min. | Typ. | Max. | Unit | Test Conditions |
|---|---|---|---|---|---|---|---|
| Power-on VCC rising gradient | At normal startup*1 | SrVCC | 0.02 | — | 20 | ms/V | |
| | During fast startup time*2 | | 0.02 | — | 2 | | |
| | Voltage monitoring 1 reset enabled at startup*3, *4 | | 0.02 | — | — | | |

Note 1. When OFS1.(STUPLVD1REN, FASTSTUP) = 11b.
Note 2. When OFS1.(STUPLVD1REN, FASTSTUP) = 10b.
Note 3. When OFS1.STUPLVD1REN = 0.
Note 4. Turn on the power supply voltage according to the normal startup rising gradient because the register settings set by OFS1 are not read in boot mode.

### Table 36.14 DC Characteristics (12)

Conditions: VCC = AVCC0 = VCC_USB = 1.8 to 3.6 V, VSS = AVSS0 = VREFL0 = VSS_USB = 0 V, $T_a$ = –40 to +105°C
The ripple voltage must meet the allowable ripple frequency $f_{r(VCC)}$ within the range between the VCC upper limit (3.6 V) and lower limit (1.8 V).
When VCC change exceeds VCC ±10%, the allowable voltage change rising/falling gradient dt/dVCC must be met.

| Item | Symbol | Min. | Typ. | Max. | Unit | Test Conditions |
|---|---|---|---|---|---|---|
| Allowable ripple frequency | $f_{r (VCC)}$ | — | — | 10 | kHz | Figure 36.6<br>$V_{r (VCC)} \leq VCC \times 0.2$ |
| | | — | — | 1 | MHz | Figure 36.6<br>$V_{r (VCC)} \leq VCC \times 0.08$ |
| | | — | — | 10 | MHz | Figure 36.6<br>$V_{r (VCC)} \leq VCC \times 0.06$ |
| Allowable voltage change rising/ falling gradient | dt/dVCC | 1.0 | — | — | ms/V | When VCC change exceeds VCC ±10% |

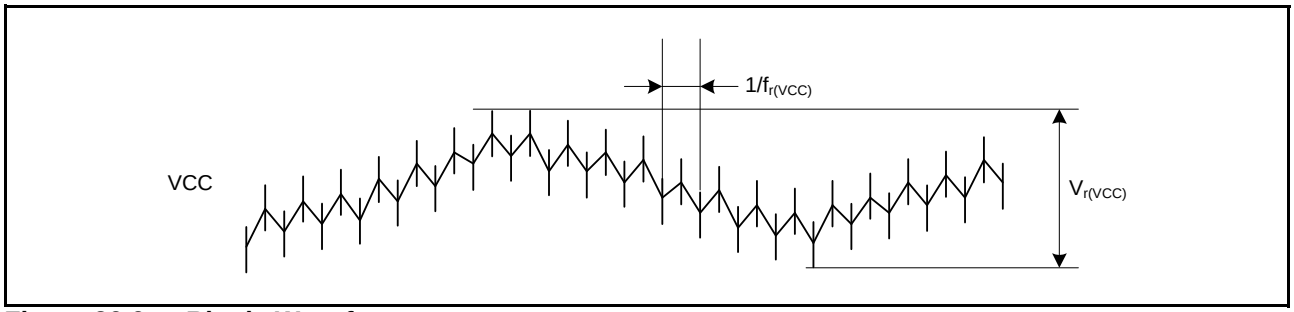



**Figure 36.6 Ripple Waveform**

### Table 36.15 DC Characteristics (13)

Conditions: VCC = AVCC0 = VCC_USB = 1.8 to 3.6 V, VSS = AVSS0 = VREFL0 = VSS_USB = 0 V, $T_a$ = –40 to +105°C

| Item | Symbol | Min. | Typ. | Max. | Unit | Test Conditions |
|---|---|---|---|---|---|---|
| Permissible error of VCL pin external capacitance | $C_{VCL}$ | 1.4 | 4.7 | 7.0 | µF | |

Note: • The recommended capacitance is 4.7 µF. Variations in connected capacitors should be within the above range.

### Table 36.16 Permissible Output Currents

Conditions: VCC = AVCC0 = VCC_USB = 1.8 to 3.6 V, VSS = AVSS0 = VREFL0 = VSS_USB = 0 V, $T_a$ = –40 to +105°C

| Item | | Symbol | Max. | Unit |
|---|---|---|---|---|
| Permissible output low current (average value per pin) | Ports 40 to 44, 46, ports J6, J7 | $I_{OL}$ | 0.4 | mA |
| | Ports other than above | | 8.0 | |
| Permissible output low current (maximum value per pin) | Ports 40 to 44, 46, ports J6, J7 | | 0.4 | |
| | Ports other than above | | 8.0 | |
| Permissible output low current | Total of ports 40 to 44, 46, ports J6, J7 | $\Sigma I_{OL}$ | 2.4 | |
| | Total of ports 03, 05, ports 26, 27, ports 30, 31 | | 30 | |
| | Total of ports 14 to 17, port 32, ports 54, 55, ports C2 to C7, ports B0, B1, B3, B5 to B7 | | 30 | |
| | Total of ports A0, A1, A3, A4, A6, port E | | 30 | |
| | Total of all output pins | | 60 | |
| Permissible output high current (average value per pin) | Ports 40 to 44, 46, ports J6, J7 | $I_{OH}$ | –0.1 | |
| | Ports other than above | | –4.0 | |
| Permissible output high current (maximum value per pin) | Ports 40 to 44, 46, ports J6, J7 | | –0.1 | |
| | Ports other than above | | –4.0 | |
| Permissible output high current | Total of ports 40 to 44, 46, ports J6, J7 | $\Sigma I_{OH}$ | –0.6 | |
| | Total of ports 03, 05, ports 26, 27, ports 30, 31 | | –10 | |
| | Total of ports 14, 15, 16, 17, port 32, ports 54, 55, ports C2 to C7, ports B0, B1, B3, B5 to B7 | | –15 | |
| | Total of ports A0, A1, A3, A4, A6, port E | | –15 | |
| | Total of all output pins | | –40 | |

Note: • Do not exceed the permissible total supply current.

### Table 36.17 Output Values of Voltage (1)

Conditions: VCC = AVCC0 = VCC_USB = 2.7 to 3.6 V, VSS = AVSS0 = VREFL0 = VSS_USB = 0 V, $T_a$ = –40 to +10°C

| Item | | | Symbol | Min. | Max. | Unit | Test Conditions |
|---|---|---|---|---|---|---|---|
| Output low | All output ports (except for RIIC, port 4, and port J) | | $V_{OL}$ | — | 0.6 | V | $I_{OL}$ = 3.0 mA |
| | | | | — | 0.4 | | $I_{OL}$ = 1.5 mA |
| | Ports 40 to 44, 46, ports J6, J7 | | | — | 0.4 | | $I_{OL}$ = 0.4 mA |
| | RIIC pins | Standard mode | | — | 0.4 | | $I_{OL}$ = 3.0 mA |
| | | Fast mode | | — | 0.6 | | $I_{OL}$ = 6.0 mA |
| Output high | All output ports (except for port 4 and port J) | | $V_{OH}$ | VCC – 0.5 | — | V | $I_{OH}$ = –2.0 mA |
| | Ports 40 to 44, 46, ports J6, J7 | | | VCC – 0.5 | — | | $I_{OH}$ = –0.1 mA |

### Table 36.18 Output Values of Voltage (2)

Conditions: VCC = AVCC0 = VCC_USB = 1.8 to 2.7 V, VSS = AVSS0 = VREFL0 = VSS_USB = 0 V, $T_a$ = –40 to +105°C

| Item | | Symbol | Min. | Max. | Unit | Test Conditions |
|---|---|---|---|---|---|---|
| Output low | All output ports (except for port 4 and port J) | $V_{OL}$ | — | 0.6 | V | $I_{OL}$ = 1.5 mA |
| | Ports 40 to 44, 46, ports J6, J7 | | — | 0.4 | | $I_{OL}$ = 0.4 mA |
| Output high | All output ports (except for port 4 and port J) | $V_{OH}$ | VCC – 0.5 | — | V | $I_{OH}$ = –1.0 mA |
| | Ports 40 to 44, 46, ports J6, J7 | | VCC – 0.5 | — | | $I_{OH}$ = –0.1 mA |

### 36.2.1 Standard I/O Pin Output Characteristics (1)

Figure 36.7 to Figure 36.10 show the characteristics of general ports (except for the RIIC output pin, port 4, and port J).

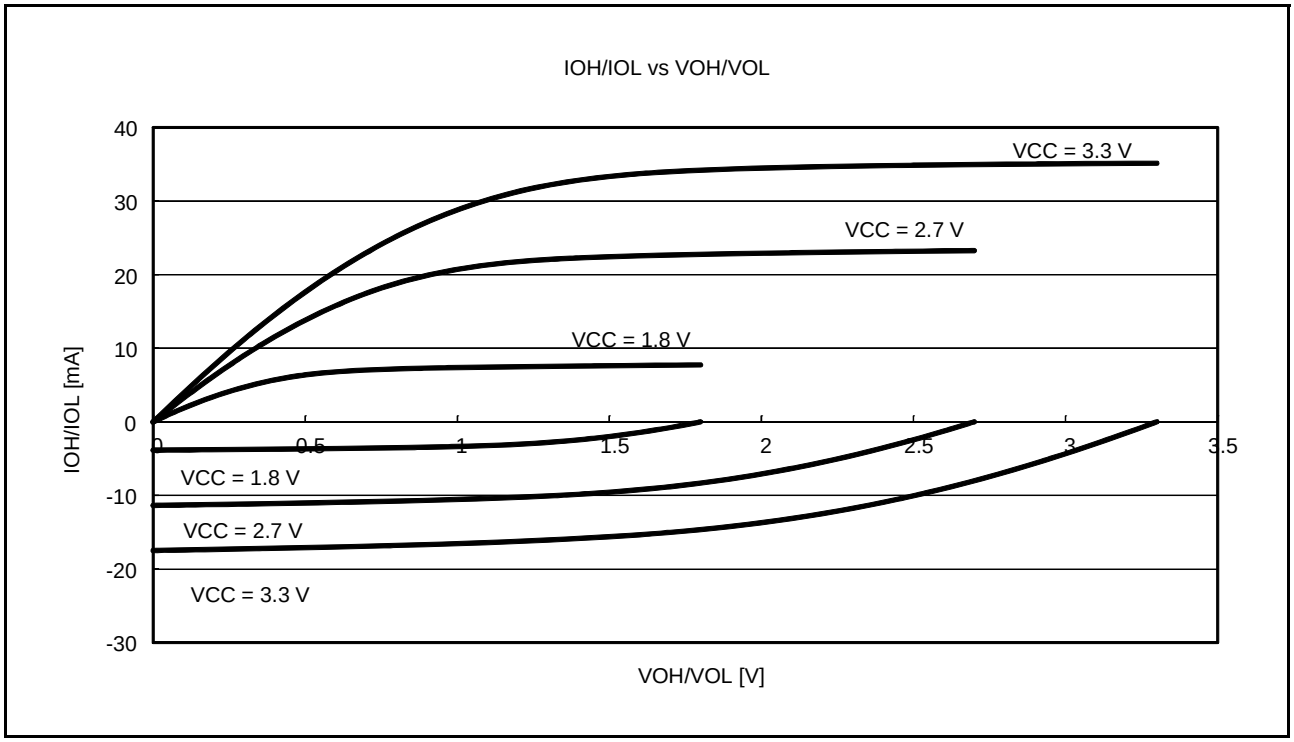



**Figure 36.7 VOH/VOL and IOH/IOL Voltage Characteristics of General Ports (Except for RIIC Output Pin, Port 4, and Port J) at $T_a$ = 25°C (Reference Data)**

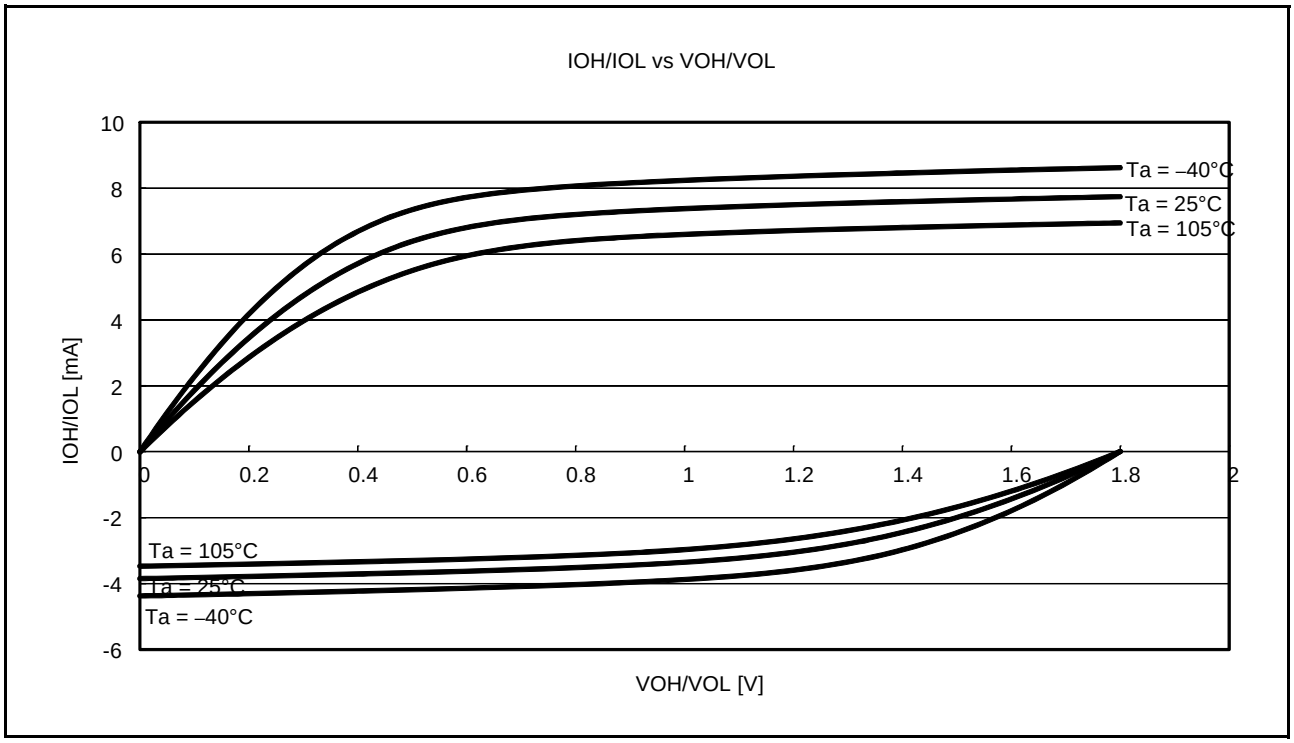



**Figure 36.8 VOH/VOL and IOH/IOL Temperature Characteristics of General Ports (Except for RIIC Output Pin, Port 4, and Port J) at VCC = 1.8 V (Reference Data)**

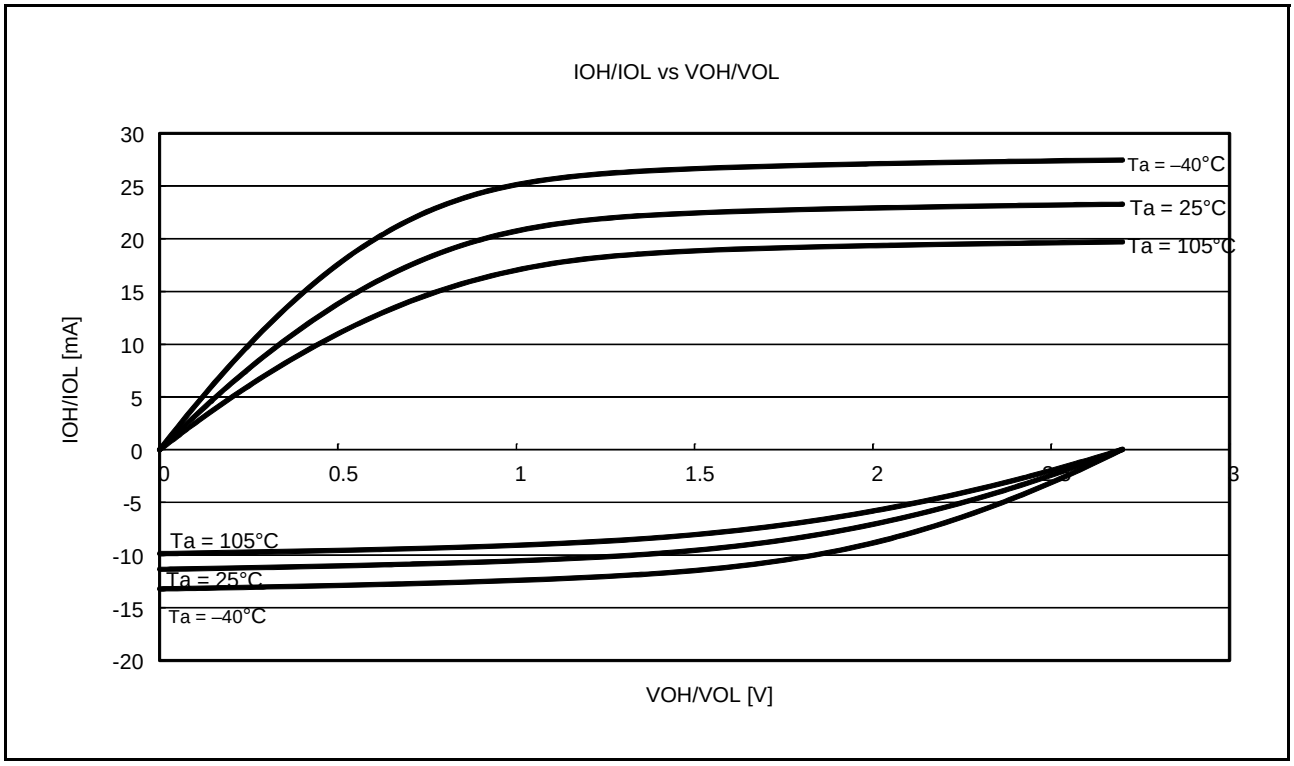



**Figure 36.9 VOH/VOL and IOH/IOL Temperature Characteristics of General Ports (Except for RIIC Output Pin, Port 4, and Port J) at VCC = 2.7 V (Reference Data)**

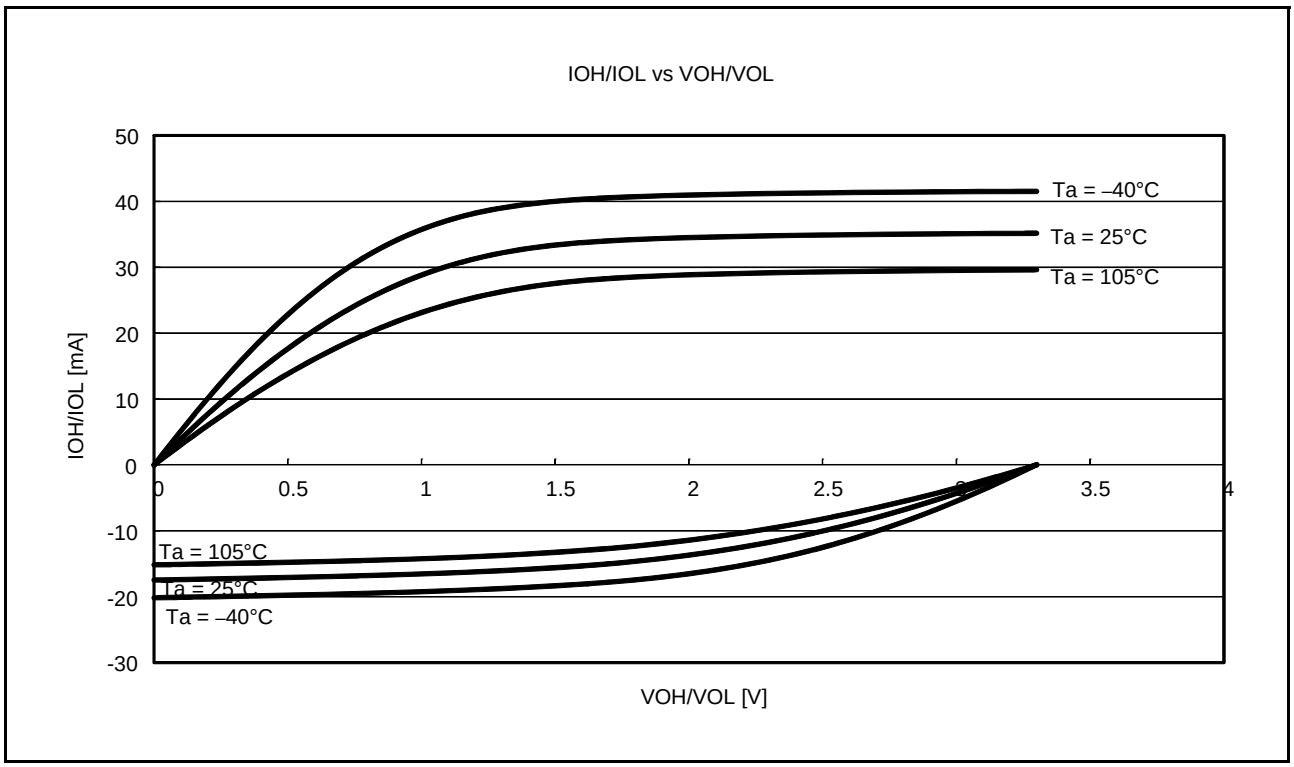



**Figure 36.10 VOH/VOL and IOH/IOL Temperature Characteristics of General Ports (Except for RIIC Output Pin, Port 4, and Port J) at VCC = 3.3 V (Reference Data)**

### 36.2.2 Standard I/O Pin Output Characteristics (2)

Figure 36.11 to Figure 36.13 show the characteristics of the RIIC output pin.

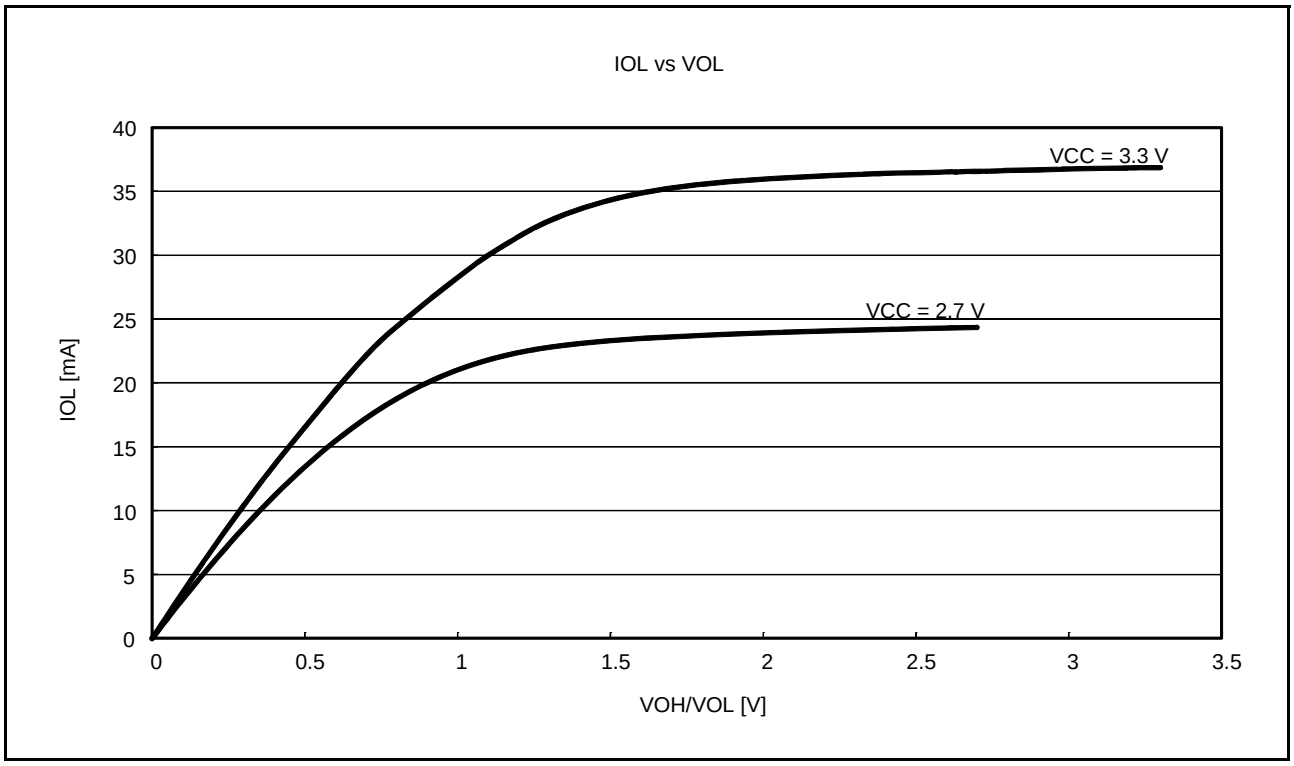



**Figure 36.11 VOL and IOL Voltage Characteristics of RIIC Output Pin at $T_a$ = 25°C (Reference Data)**

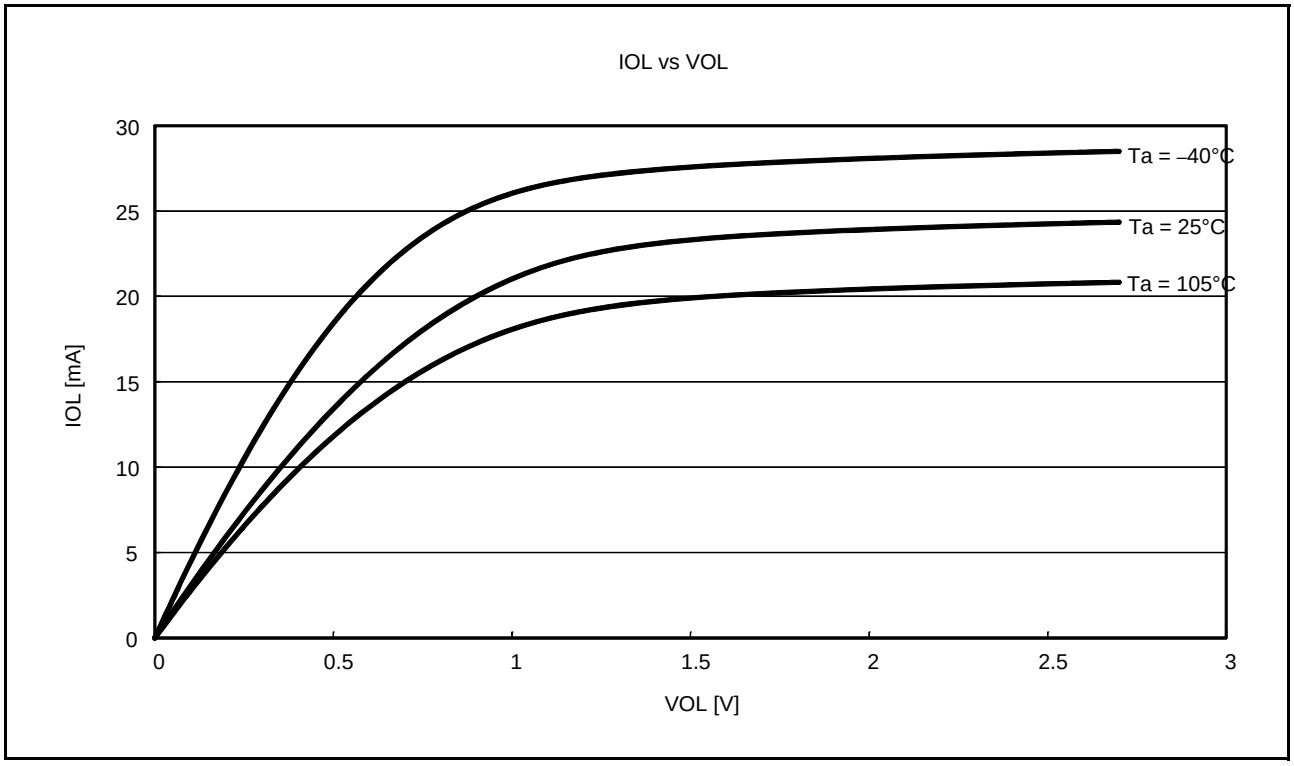



**Figure 36.12 VOL and IOL Temperature Characteristics of RIIC Output Pin at VCC = 2.7 V (Reference Data)**

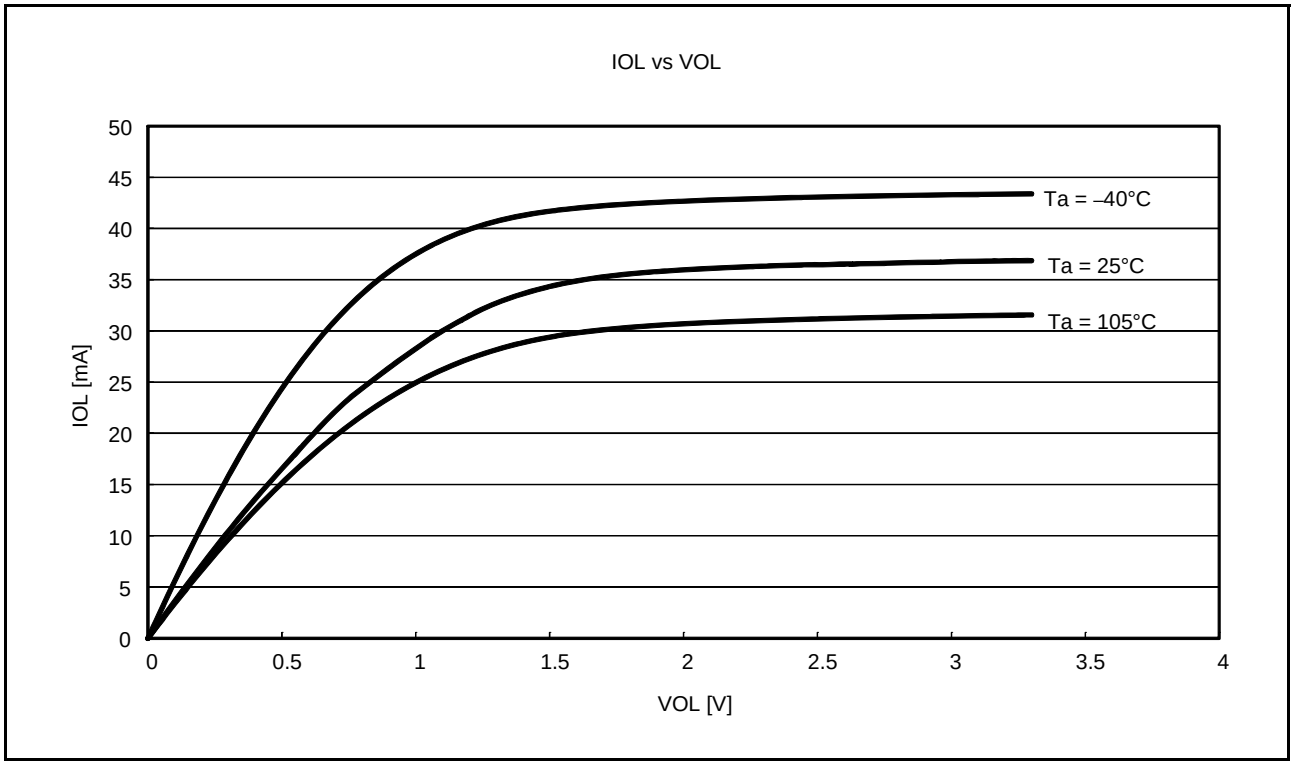



**Figure 36.13 VOL and IOL Temperature Characteristics of RIIC Output Pin at VCC = 3.3 V (Reference Data)**

### 36.2.3 Standard I/O Pin Output Characteristics (3)

Figure 36.14 to Figure 36.17 show the characteristics of port 4 and port J.

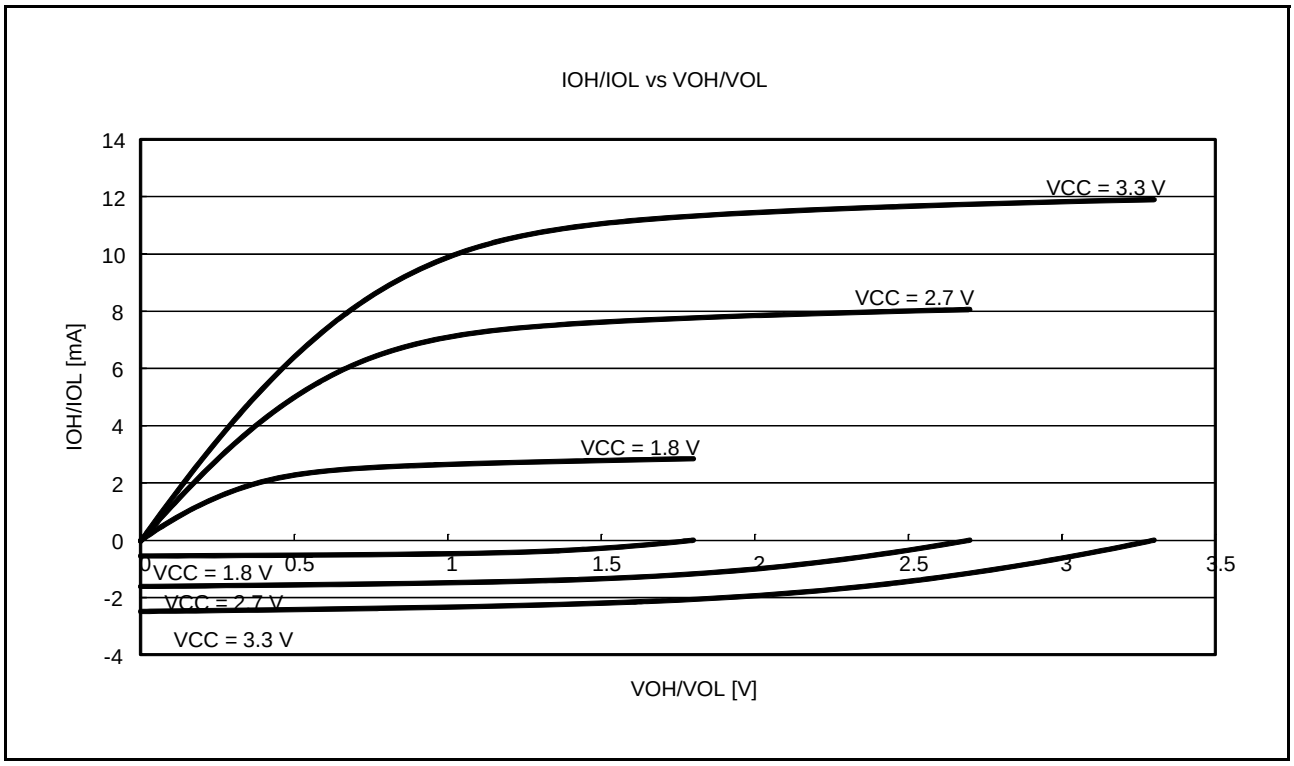



**Figure 36.14 VOH/VOL and IOH/IOL Voltage Characteristics of Port 4 and Port J at $T_a$ = 25°C (Reference Data)**

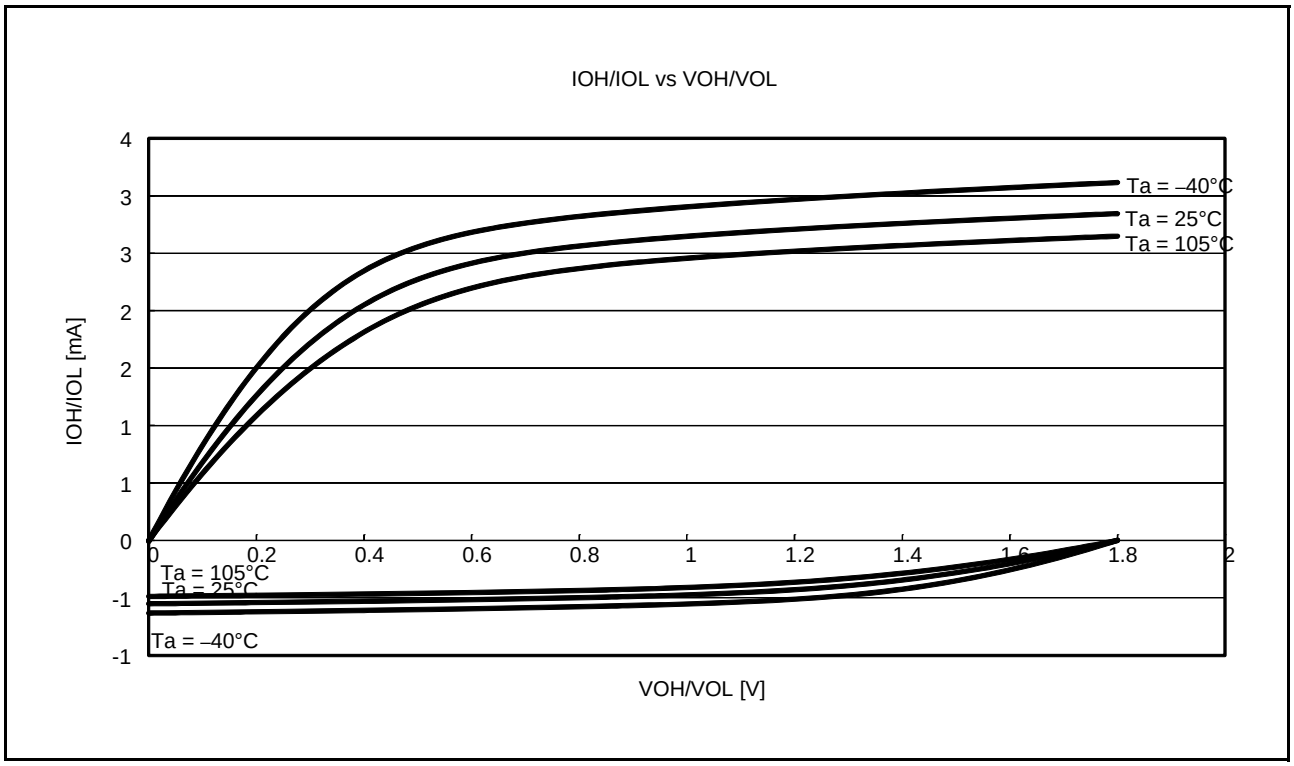



**Figure 36.15 VOH/VOL and IOH/IOL Temperature Characteristics of Port 4 and Port J at VCC = 1.8 V (Reference Data)**

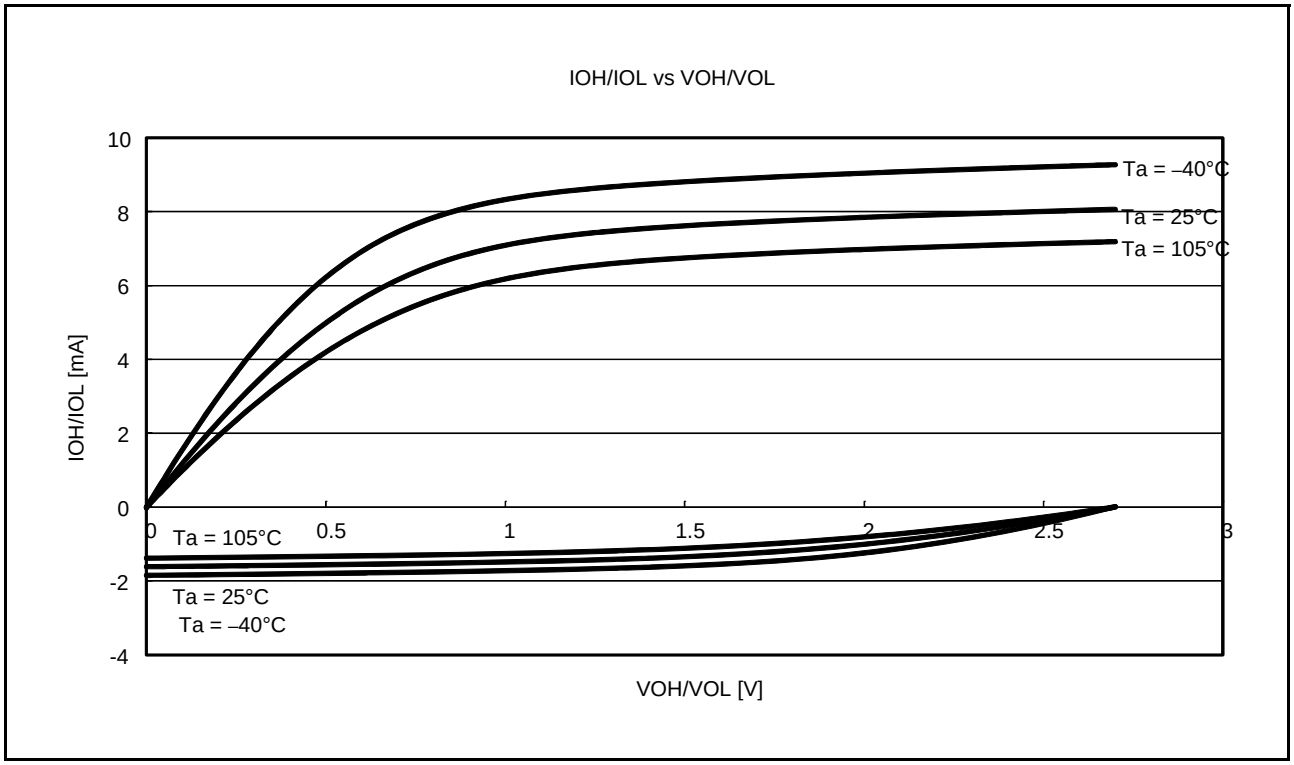



**Figure 36.16 VOH/VOL and IOH/IOL Temperature Characteristics of Port 4 and Port J at VCC = 2.7 V (Reference Data)**

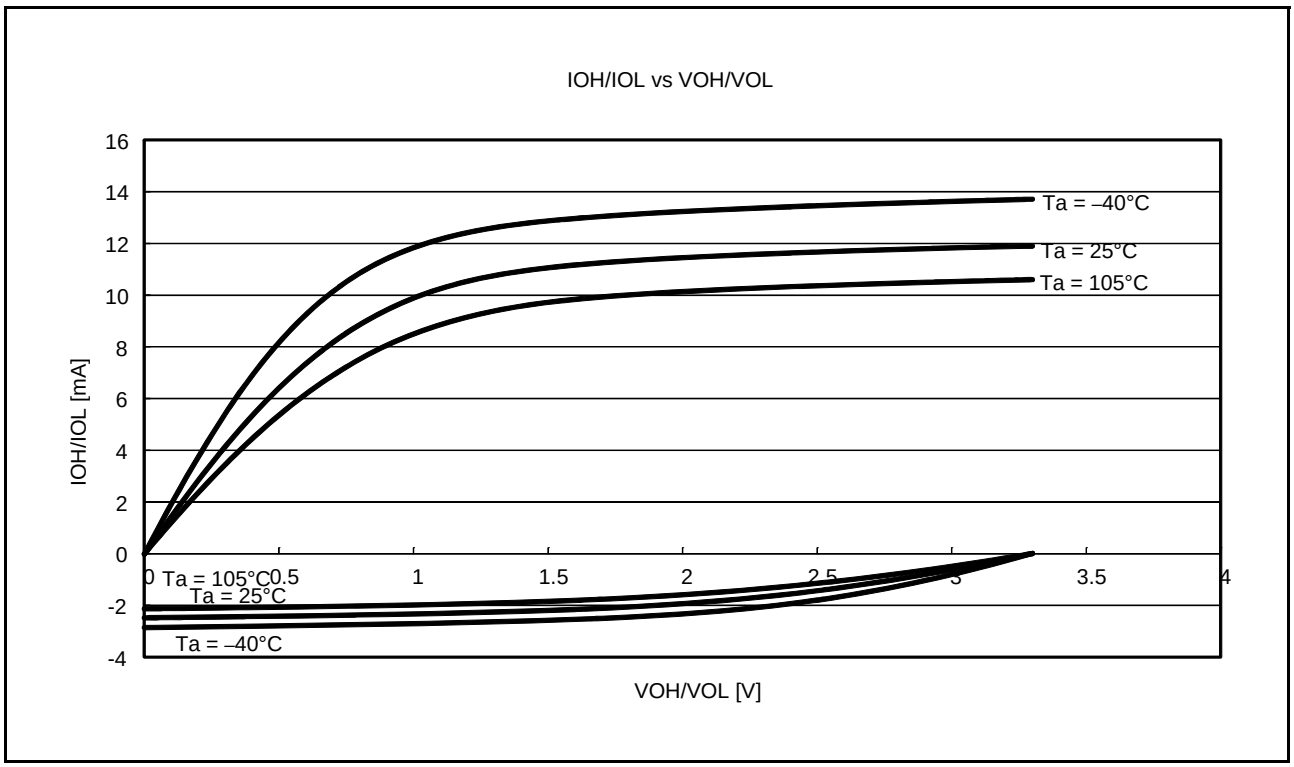



**Figure 36.17 VOH/VOL and IOH/IOL Temperature Characteristics of Port 4 and Port J at VCC = 3.3 V (Reference Data)**

## 36.3 AC Characteristics

### 36.3.1 Clock Timing

**Table 36.19 Operation Frequency Value (High-Speed Operating Mode)**

Conditions: VCC = AVCC0 = VCC_USB = 1.8 to 3.6 V, VSS = AVSS0 = VREFL0 = VSS_USB = 0 V, $T_a$ = –40 to +105°C

| Item | | Symbol | VCC | | | | Unit |
|---|---|---|---|---|---|---|---|
| | | | 1.8 to 2.4 V | 2.4 to 2.7 V | 2.7 to 3.6 V | When USB in Use*4 | |
| Maximum operating frequency | System clock (ICLK) | $f_{max}$ | 8 | 16 | 32 | 24 | MHz |
| | FlashIF clock (FCLK)*1, *2 | | 8 | 16 | 32 | 24 | |
| | Peripheral module clock (PCLKB) | | 8 | 16 | 32 | 24 | |
| | Peripheral module clock (PCLKD)*3 | | 8 | 16 | 32 | 24 | |
| | USB clock (UCLK) | $f_{usb}$ | — | — | — | 48 | |

Note 1. The lower-limit frequency of FCLK is 1 MHz during programming or erasing of the flash memory. When using FCLK at below 4 MHz, the frequency can be set to 1 MHz, 2 MHz, or 3 MHz. A non-integer frequency such as 1.5 MHz cannot be set.
Note 2. The frequency accuracy of FCLK should be ±3.5%. Confirm the frequency accuracy of the clock source.
Note 3. The lower-limit frequency of PCLKD is 4 MHz at 2.4 V or above and 1 MHz at below 2.4 V when the A/D converter is in use.
Note 4. The VCC_USB range is 3.0 to 3.6 V when the USB clock is in use.

**Table 36.20 Operation Frequency Value (Middle-Speed Operating Mode)**

Conditions: VCC = AVCC0 = VCC_USB = 1.8 to 3.6 V, VSS = AVSS0 = VREFL0 = VSS_USB = 0 V, $T_a$ = –40 to +105°C

| Item | | Symbol | VCC | | | Unit |
|---|---|---|---|---|---|---|
| | | | 1.8 to 2.4 V | 2.4 to 2.7 V | 2.7 to 3.6 V | |
| Maximum operating frequency | System clock (ICLK) | $f_{max}$ | 8 | 12 | 12 | MHz |
| | FlashIF clock (FCLK)*1, *2 | | 8 | 12 | 12 | |
| | Peripheral module clock (PCLKB) | | 8 | 12 | 12 | |
| | Peripheral module clock (PCLKD)*3 | | 8 | 12 | 12 | |

Note 1. The lower-limit frequency of FCLK is 1 MHz during programming or erasing of the flash memory. When using FCLK at below 4 MHz, the frequency can be set to 1 MHz, 2 MHz, or 3 MHz. A non-integer frequency such as 1.5 MHz cannot be set.
Note 2. The frequency accuracy of FCLK should be ±3.5%.
Note 3. The lower-limit frequency of PCLKD is 4 MHz at 2.4 V or above and 1 MHz at below 2.4 V when the A/D converter is in use.

**Table 36.21 Operation Frequency Value (Low-Speed Operating Mode)**

Conditions: VCC = AVCC0 = VCC_USB = 1.8 to 3.6 V, VSS = AVSS0 = VREFL0 = VSS_USB = 0 V, $T_a$ = –40 to +105°C

| Item | | Symbol | VCC | | | Unit |
|---|---|---|---|---|---|---|
| | | | 1.8 to 2.4 V | 2.4 to 2.7 V | 2.7 to 3.6 V | |
| Maximum operating frequency | System clock (ICLK) | $f_{max}$ | 32.768 | | | kHz |
| | FlashIF clock (FCLK)*1 | | 32.768 | | | |
| | Peripheral module clock (PCLKB) | | 32.768 | | | |
| | Peripheral module clock (PCLKD)*2 | | 32.768 | | | |

Note 1. Programming and erasing the flash memory is impossible.
Note 2. The A/D converter cannot be used.

**Table 36.22 Clock Timing**

Conditions: VCC = AVCC0 = VCC_USB = 1.8 to 3.6 V, VSS = AVSS0 = VREFL0 = VSS_USB = 0 V, $T_a$ = –40 to +105°C

| Item | | Symbol | Min. | Typ. | Max. | Unit | Test Conditions |
|---|---|---|---|---|---|---|---|
| XTAL external clock input cycle time | | $t_{Xcyc}$ | 50 | — | — | ns | Figure 36.18 |
| XTAL external clock input high pulse width | | $t_{XH}$ | 20 | — | — | ns | |
| XTAL external clock input low pulse width | | $t_{XL}$ | 20 | — | — | ns | |
| XTAL external clock rising time | | $t_{Xr}$ | — | — | 5 | ns | |
| XTAL external clock falling time | | $t_{Xf}$ | — | — | 5 | ns | |
| XTAL external clock input wait time*1 | | $t_{EXWT}$ | 0.5 | — | — | μs | |
| Main clock oscillator oscillation frequency*2 | 2.4 ≤ VCC ≤ 3.6 | $f_{MAIN}$ | 1 | — | 20 | MHz | |
| | 1.8 ≤ VCC < 2.4 | | 1 | — | 8 | | |
| Main clock oscillation stabilization time (crystal)*2 | | $t_{MAINOSC}$ | — | 3 | — | ms | Figure 36.20 |
| Main clock oscillation stabilization time (ceramic resonator)*2 | | $t_{MAINOSC}$ | — | 50 | | μs | |
| LOCO clock oscillation frequency | | $f_{LOCO}$ | 3.44 | 4.0 | 4.56 | MHz | |
| LOCO clock oscillation stabilization time | | $t_{LOCO}$ | — | — | 0.5 | μs | Figure 36.21 |
| IWDT-dedicated clock oscillation frequency | | $f_{ILOCO}$ | 12.75 | 15 | 17.25 | kHz | |
| IWDT-dedicated clock oscillation stabilization time | | $t_{ILOCO}$ | — | — | 50 | μs | Figure 36.19 |
| HOCO clock oscillation frequency | | $f_{HOCO}$ | 31.52 | 32 | 32.48 | MHz | Ta = –40 to 85°C |
| | | | 31.68 | 32 | 32.32 | | Ta = –20 to 85°C |
| | | | 31.36 | 32 | 32.64 | | Ta = –40 to 105°C |
| HOCO clock oscillation stabilization time | | $t_{HOCO2}$ | — | — | 56 | μs | Figure 36.23 |
| PLL input frequency*3 | | $f_{PLLIN}$ | 4 | — | 8 | MHz | |
| PLL circuit oscillation frequency*3 | | $f_{PLL}$ | 32 | — | 48 | MHz | |
| PLL clock oscillation stabilization time | | $t_{PLL}$ | — | — | 50 | μs | Figure 36.24 |
| Sub-clock oscillator oscillation frequency | | $f_{SUB}$ | — | 32.768 | — | kHz | |
| Sub-clock oscillation stabilization time*4 | | $t_{SUBOSC}$ | — | 0.5 | — | s | Figure 36.25 |

Note 1. Time until the clock can be used after the main clock oscillator stop bit (MOSCCR.MOSTP) is set to 0 (operating) when the external clock is stable.

Note 2. Reference values when an 8-MHz oscillator is used.
When specifying the main clock oscillator stabilization time, set the MOSCWTCR register with a stabilization time value that is equal to or greater than the oscillator-manufacturer-recommended value.
After changing the setting of the MOSCCR.MOSTP bit so that the main clock oscillator operates, read the OSCOVFSR.MOOVF flag to confirm that is has become 1, and then start using the main clock.

Note 3. The VCC range that the PLL can be used is 2.4 to 3.6 V.

Note 4. After changing the setting of the SOSCCR.SOSTP bit or RCR3.RTCEN bit so that the sub-clock oscillator operates, only start using the sub-clock after the sub-clock oscillation stabilization time with an adequate margin (2 times is recommended) has elapsed.

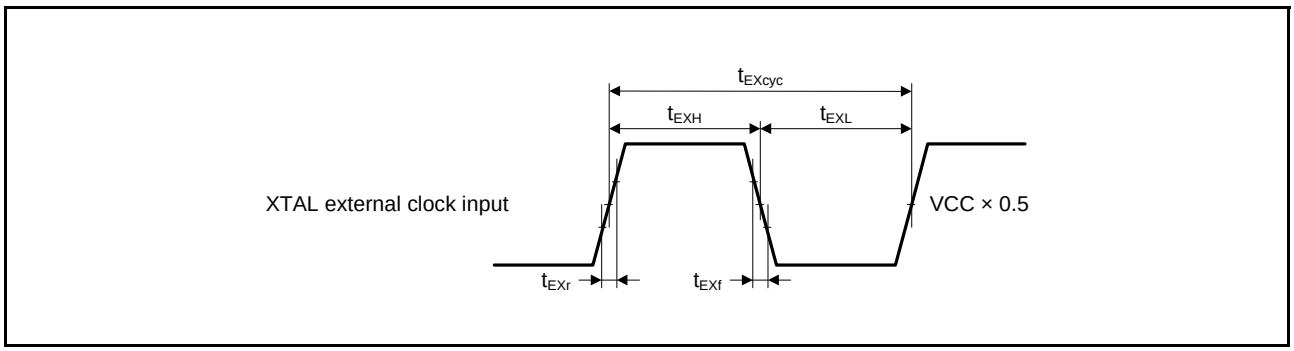



**Figure 36.18 XTAL External Clock Input Timing**

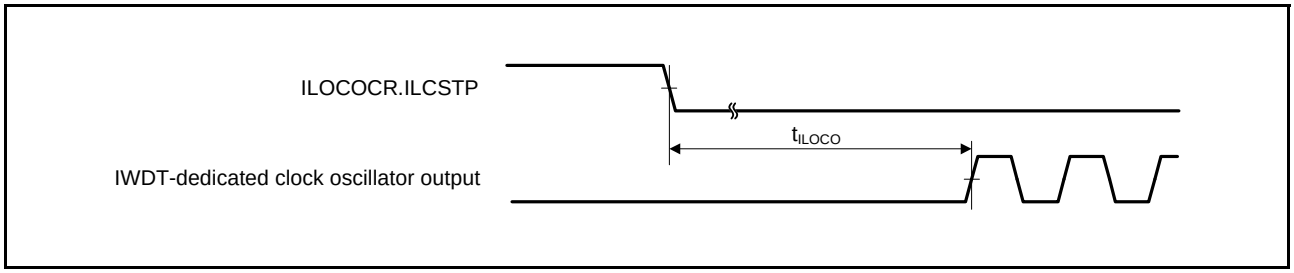



**Figure 36.19 IWDT-Dedicated Clock Oscillation Start Timing**

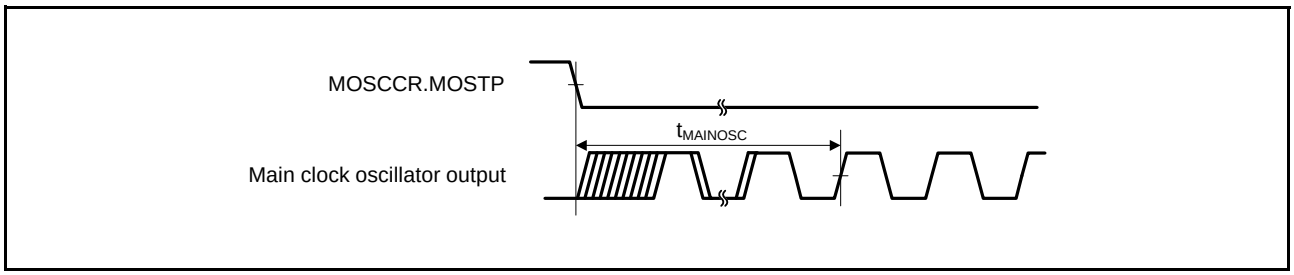



**Figure 36.20 Main Clock Oscillation Start Timing**

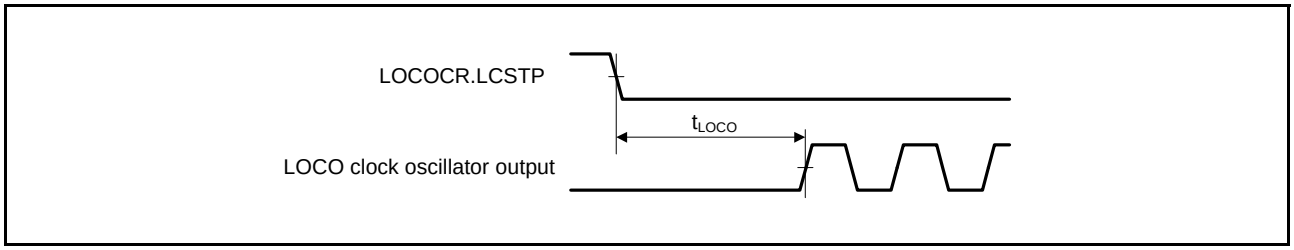



**Figure 36.21 LOCO Clock Oscillation Start Timing**

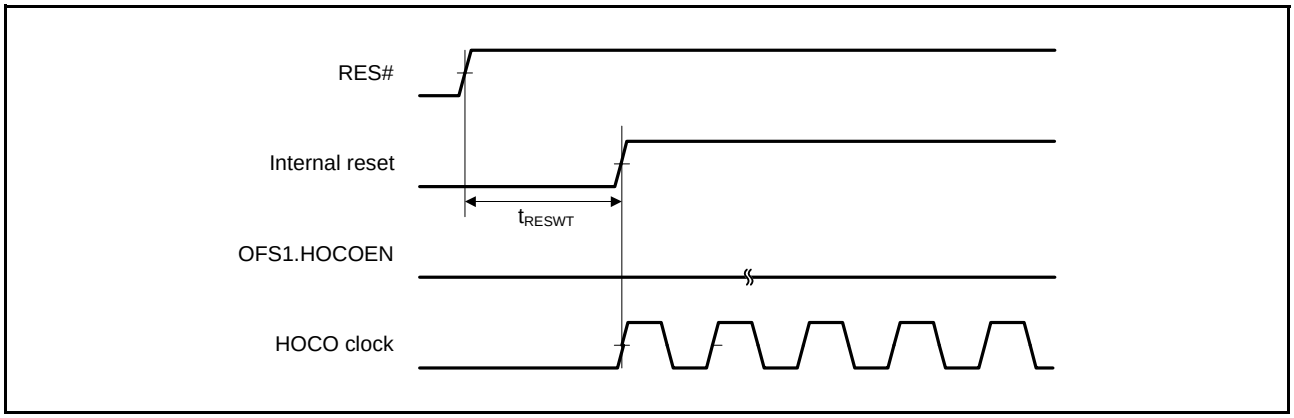



**Figure 36.22 HOCO Clock Oscillation Start Timing (After Reset is Canceled by Setting OFS1.HOCOEN Bit to 0)**

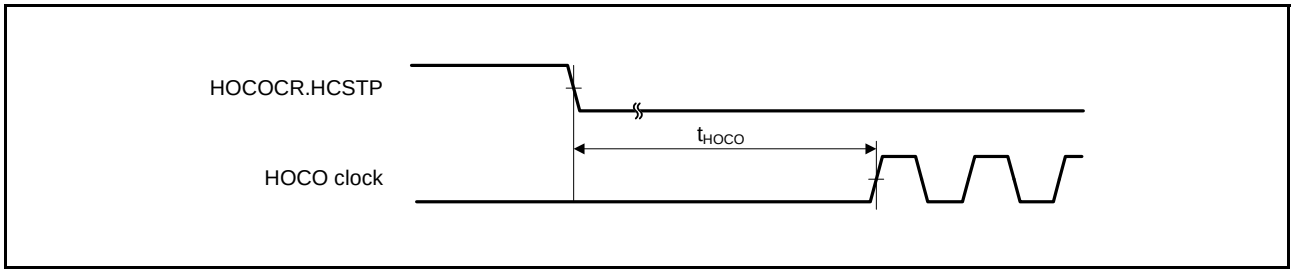



**Figure 36.23 HOCO Clock Oscillation Start Timing (Oscillation is Started by Setting HOCOCR.HCSTP Bit)**

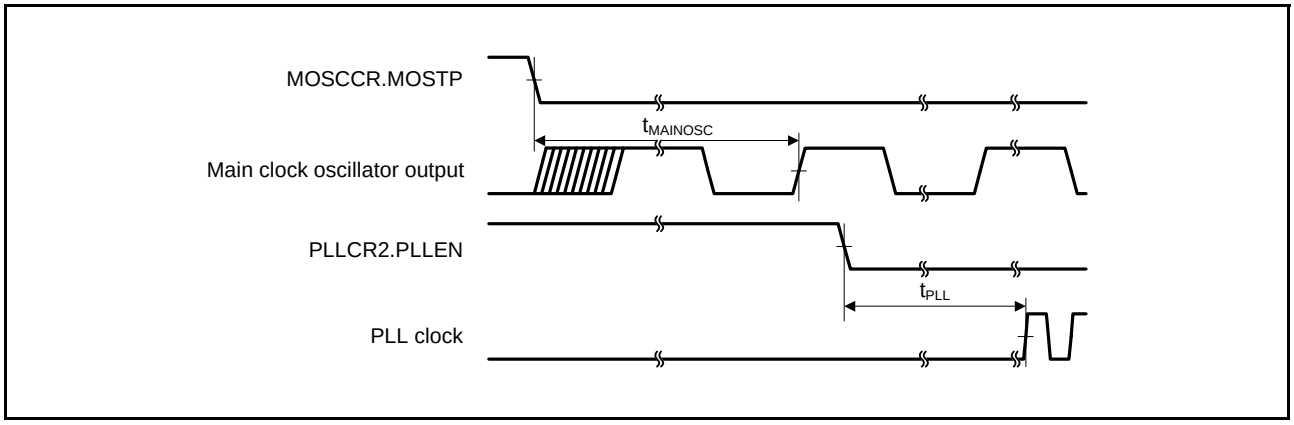



**Figure 36.24 PLL Clock Oscillation Start Timing (PLL is Operated after Main Clock Oscillation Has Settled)**

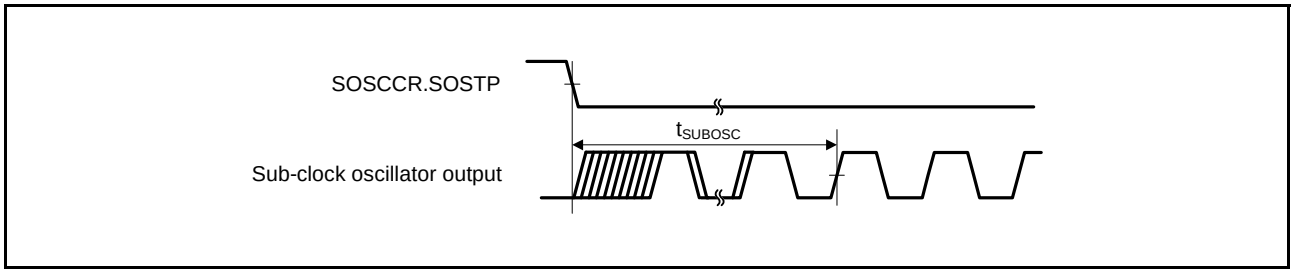



**Figure 36.25 Sub-Clock Oscillation Start Timing**

## 36.3.2 Reset Timing

**Table 36.23 Reset Timing**

Conditions: VCC = AVCC0 = VCC_USB = 1.8 to 3.6 V, VSS = AVSS0 = VREFL0 = VSS_USB = 0 V, $T_a$ = –40 to +105°C

| Item | | Symbol | Min. | Typ. | Max. | Unit | Test Conditions |
|---|---|---|---|---|---|---|---|
| RES# pulse width | At power-on | $t_{RESWP}$ | 3 | — | — | ms | Figure 36.26 |
| | Other than above | $t_{RESW}$ | 30 | — | — | µs | Figure 36.27 |
| Wait time after RES# cancellation (at power-on) | At normal startup*1 | $t_{RESWT}$ | — | 8.5 | — | ms | Figure 36.26 |
| | During fast startup time*2 | $t_{RESWT}$ | — | 560 | — | µs | |
| Wait time after RES# cancellation (during powered-on state) | | $t_{RESWT}$ | — | 114 | — | µs | Figure 36.27 |
| Independent watchdog timer reset period | | $t_{RESWIW}$ | — | 1 | — | IWDT clock cycle | Figure 36.28 |
| Software reset period | | $t_{RESWSW}$ | — | 1 | — | ICLK cycle | |
| Wait time after independent watchdog timer reset cancellation*3 | | $t_{RESW2}$ | — | 300 | — | µs | |
| Wait time after software reset cancellation | | $t_{RESW2}$ | — | 168 | — | µs | |

Note 1. When OFS1.(STUPLVD1REN, FASTSTUP) = 11b.
Note 2. When OFS1.(STUPLVD1REN, FASTSTUP) ≠ 11b.
Note 3. When IWDTCR.CKS[3:0] = 0000b.

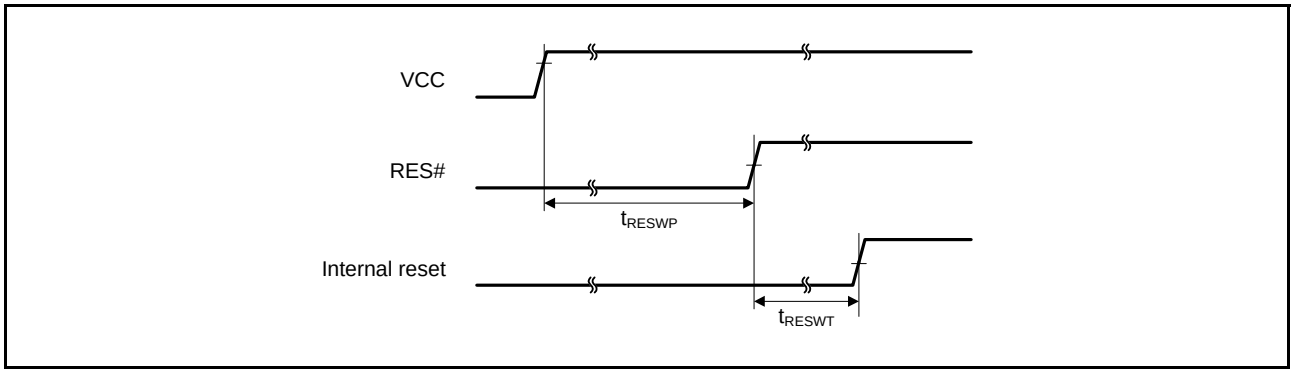



**Figure 36.26 Reset Input Timing at Power-On**

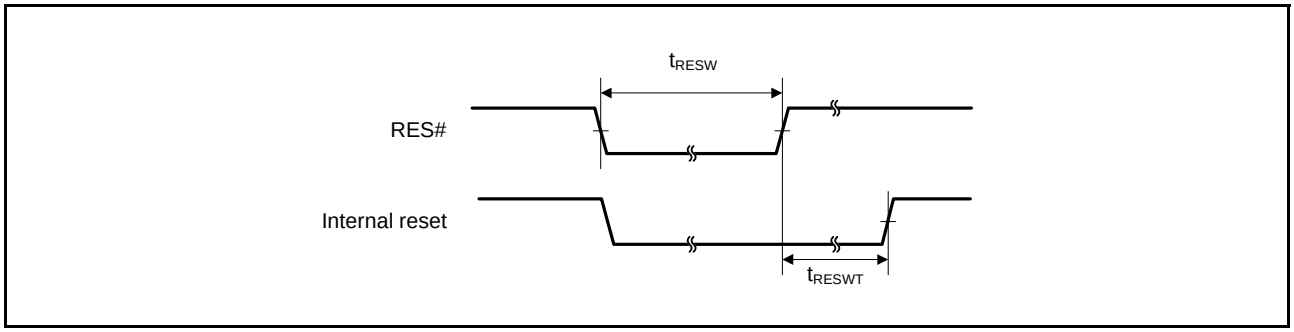



**Figure 36.27 Reset Input Timing (1)**

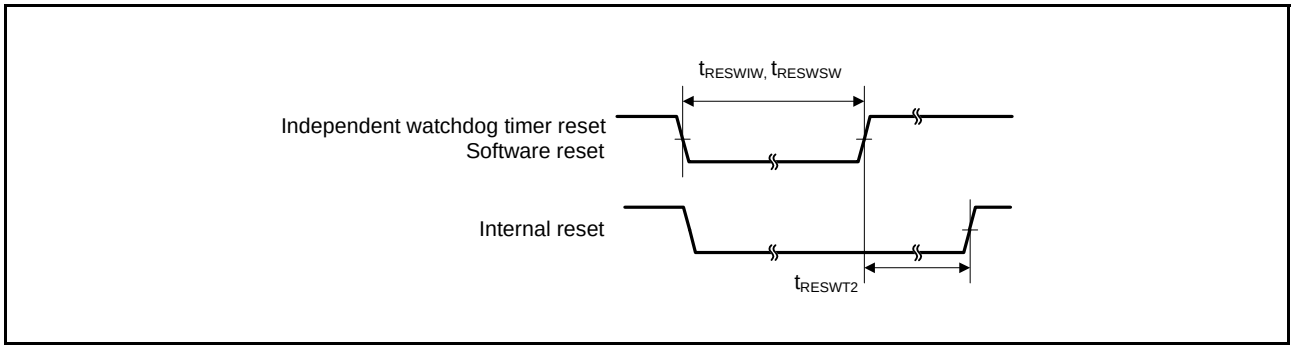



**Figure 36.28 Reset Input Timing (2)**

## 36.3.3 Timing of Recovery from Low Power Consumption Modes

**Table 36.24 Timing of Recovery from Low Power Consumption Modes (1)**

Conditions: VCC = AVCC0 = VCC_USB = 1.8 to 3.6 V, VSS = AVSS0 = VREFL0 = VSS_USB = 0 V, $T_a$ = –40 to +105°C

| Item | | | | Symbol | Min. | Typ. | Max. | Unit | Test Conditions |
|---|---|---|---|---|---|---|---|---|---|
| Recovery time from software standby mode*1 | High-speed mode | Crystal connected to main clock oscillator | Main clock oscillator operating*2 | $t_{SBYMC}$ | — | 2 | 3 | ms | Figure 36.29 |
| | | | Main clock oscillator and PLL circuit operating*3 | $t_{SBYPC}$ | — | 2 | 3 | ms | |
| | | External clock input to main clock oscillator | Main clock oscillator operating*4 | $t_{SBYEX}$ | — | 35 | 50 | µs | |
| | | | Main clock oscillator and PLL circuit operating*5 | $t_{SBYPE}$ | — | 70 | 95 | µs | |
| | | Sub-clock oscillator operating | | $t_{SBYSC}$ | — | 650 | 800 | µs | |
| | | HOCO clock oscillator operating*6 | | $t_{SBYHO}$ | — | 40 | 55 | µs | |
| | | LOCO clock oscillator operating | | $t_{SBYLO}$ | — | 40 | 55 | µs | |

Note 1. The recovery time varies depending on the state of each oscillator when the WAIT instruction is executed. The recovery time when multiple oscillators are operating varies depending on the operating state of the oscillators that are not selected as the system clock source. This applies when only the oscillator listed in each item is operating and the other oscillators are stopped.
Note 2. When the frequency of the crystal is 20 MHz.
When the main clock oscillator wait control register (MOSCWTCR) is set to 04h.
Note 3. When the frequency of PLL is 32 MHz.
When the main clock oscillator wait control register (MOSCWTCR) is set to 04h.
Note 4. When the frequency of the external clock is 20 MHz.
When the main clock oscillator wait control register (MOSCWTCR) is set to 00h.
Note 5. When the frequency of PLL is 32 MHz.
When the main clock oscillator wait control register (MOSCWTCR) is set to 00h.
Note 6. When the frequency of HOCO is 32 MHz.
When the high-speed clock oscillator wait control register (HOCOWTCR) is set to 05h.

**Table 36.25 Timing of Recovery from Low Power Consumption Modes (2)**

Conditions: VCC = AVCC0 = VCC_USB = 1.8 to 3.6 V, VSS = AVSS0 = VREFL0 = VSS_USB = 0 V, $T_a$ = –40 to +105°C

| Item | | | | Symbol | Min. | Typ. | Max. | Unit | Test Conditions |
|---|---|---|---|---|---|---|---|---|---|
| Recovery time from software standby mode*1 | Middle-speed mode | Crystal connected to main clock oscillator | Main clock oscillator operating*2 | $t_{SBYMC}$ | — | 2 | 3 | ms | Figure 36.29 |
| | | | Main clock oscillator and PLL circuit operating*3 | $t_{SBYPC}$ | — | 2 | 3 | ms | |
| | | External clock input to main clock oscillator | Main clock oscillator operating*4 | $t_{SBYEX}$ | — | 3 | 4 | µs | |
| | | | Main clock oscillator and PLL circuit operating*5 | $t_{SBYPE}$ | — | 65 | 85 | µs | |
| | | Sub-clock oscillator operating | | $t_{SBYSC}$ | — | 600 | 750 | µs | |
| | | HOCO clock oscillator operating*6 | | $t_{SBYHO}$ | — | 40 | 50 | µs | |
| | | LOCO clock oscillator operating | | $t_{SBYLO}$ | — | 4.8 | 7 | µs | |

Note 1. The recovery time varies depending on the state of each oscillator when the WAIT instruction is executed. The recovery time when multiple oscillators are operating varies depending on the operating state of the oscillators that are not selected as the system clock source. This applies when only the oscillator listed in each item is operating and the other oscillators are stopped.
Note 2. When the frequency of the crystal is 12 MHz.
When the main clock oscillator wait control register (MOSCWTCR) is set to 04h.
Note 3. When the frequency of PLL is 12 MHz.
When the main clock oscillator wait control register (MOSCWTCR) is set to 04h.
Note 4. When the frequency of the external clock is 12 MHz.
When the main clock oscillator wait control register (MOSCWTCR) is set to 00h.
Note 5. When the frequency of PLL is 12 MHz.
When the main clock oscillator wait control register (MOSCWTCR) is set to 00h.
Note 6. When the frequency of HOCO is 8 MHz.
When the high-speed clock oscillator wait control register (HOCOWTCR) is set to 05h.

**Table 36.26 Timing of Recovery from Low Power Consumption Modes (3)**

Conditions: VCC = AVCC0 = VCC_USB = 1.8 to 3.6 V, VSS = AVSS0 = VREFL0 = VSS_USB = 0 V, $T_a$ = –40 to +105°C

| Item | | | Symbol | Min. | Typ. | Max. | Unit | Test Conditions |
|---|---|---|---|---|---|---|---|---|
| Recovery time from software standby mode*1 | Low-speed mode | Sub-clock oscillator operating | $t_{SBYSC}$ | — | 600 | 750 | µs | Figure 36.29 |

Note 1. The sub-clock continues oscillating in software standby mode during low-speed mode.

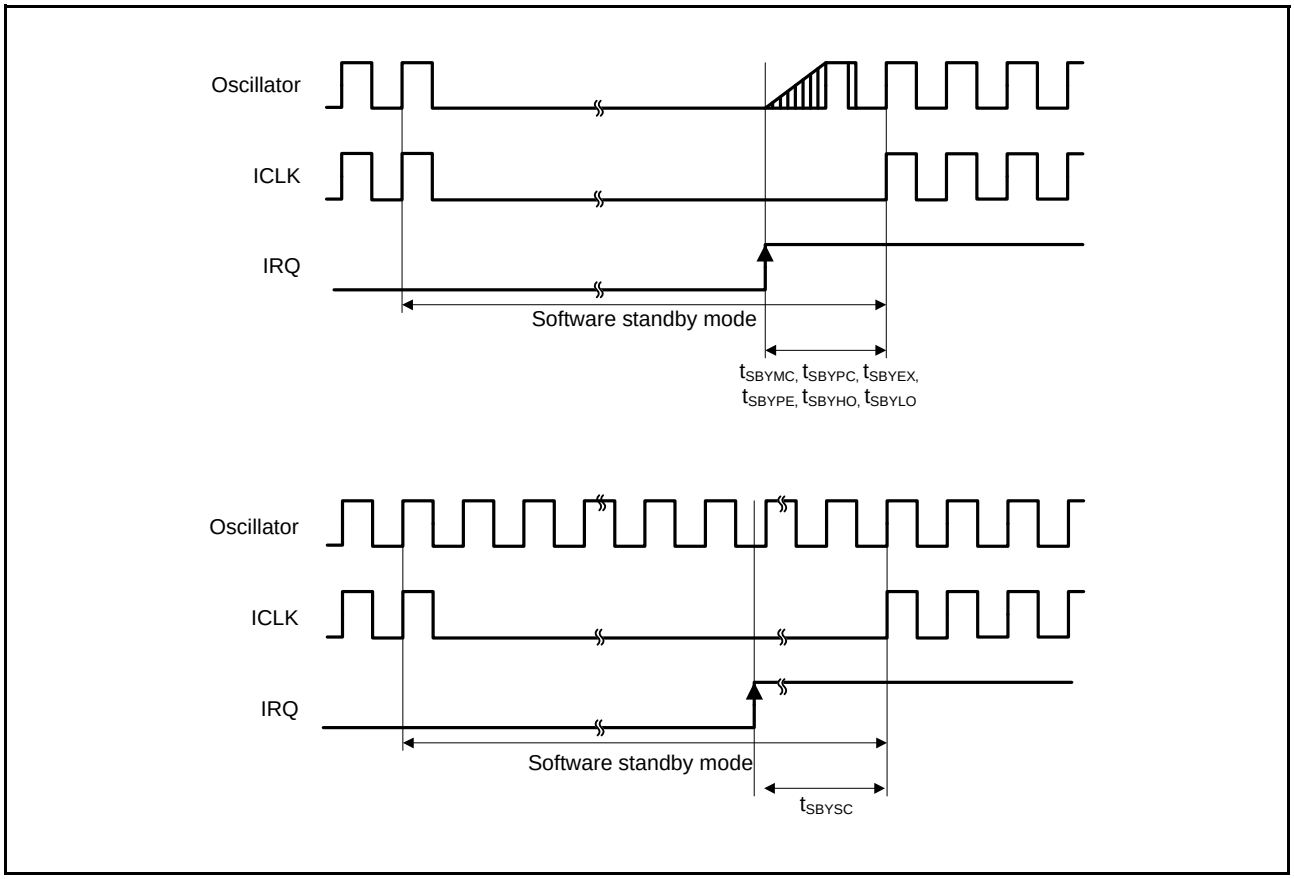



**Figure 36.29 Software Standby Mode Cancellation Timing**

**Table 36.27 Timing of Recovery from Low Power Consumption Modes (4)**

Conditions: VCC = AVCC0 = VCC_USB = 1.8 to 3.6 V, VSS = AVSS0 = VREFL0 = VSS_USB = 0 V, $T_a$ = –40 to +105°C

| Item | | Symbol | Min. | Typ. | Max. | Unit | Test Conditions |
|---|---|---|---|---|---|---|---|
| Recovery time from deep sleep mode*1 | High-speed mode*2 | $t_{DSLP}$ | — | 2 | 3.5 | µs | |
| | Middle-speed mode*3 | $t_{DSLP}$ | — | 3 | 4 | µs | |
| | Low-speed mode*4 | $t_{DSLP}$ | — | 400 | 500 | µs | |

Note 1. Oscillators continue oscillating in deep sleep mode.
Note 2. When the frequency of the system clock is 32 MHz.
Note 3. When the frequency of the system clock is 12 MHz.
Note 4. When the frequency of the system clock is 32.768 kHz.

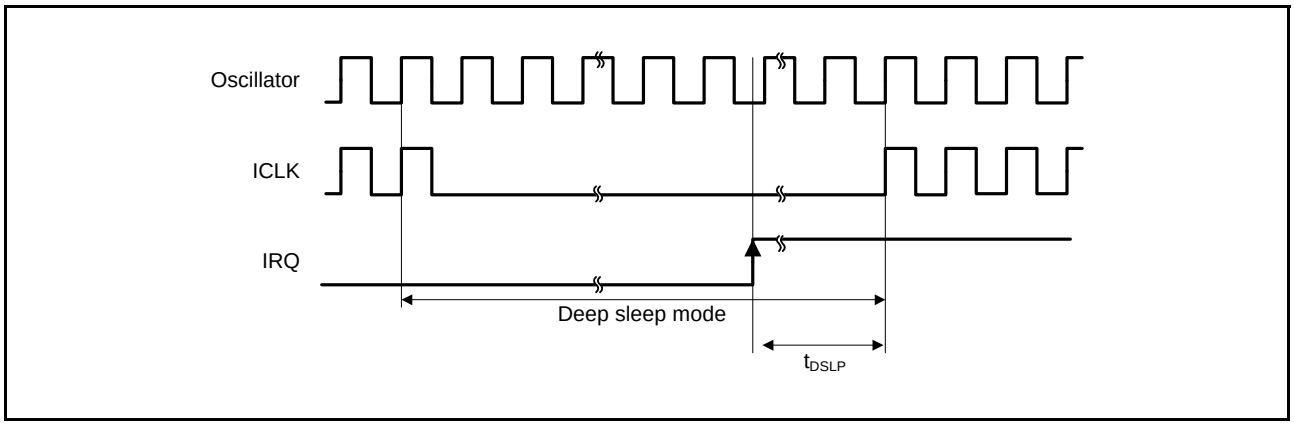



**Figure 36.30 Deep Sleep Mode Cancellation Timing**

**Table 36.28 Timing of Recovery from Low Power Consumption Modes (5) Operating Mode Transition Time**

Conditions: VCC = AVCC0 = VCC_USB = 1.8 to 3.6 V, VSS = AVSS0 = VREFL0 = VSS_USB = 0 V, $T_a$ = -40 to +105°C

| Mode before Transition | Mode after Transition | ICLK Frequency | Transition Time | | | Unit |
|---|---|---|---|---|---|---|
| | | | Min. | Typ. | Max. | |
| High-speed operating mode | Middle-speed operating mode | 8 MHz | — | 10 | — | µs |
| Middle-speed operating mode | High-speed operating mode | 8 MHz | — | 37.5 | — | µs |
| Low-speed operating mode | Middle-speed operating mode, high-speed operating mode | 32.768 kHz | — | 213.62 | — | µs |
| Middle-speed operating mode, high-speed operating mode | Low-speed operating mode | 32.768 kHz | — | 183.11 | — | µs |

Note: • When PCLKB, PCLKD, and FCLK are set to the same frequency division ratio as ICLK.

## 36.3.4 Control Signal Timing

**Table 36.29 Control Signal Timing**

Conditions: VCC = AVCC0 = VCC_USB = 1.8 to 3.6 V, VSS = AVSS0 = VREFL0 = VSS_USB = 0 V, $T_a$ = –40 to +105°C

| Item | Symbol | Min. | Typ. | Max. | Unit | Test Conditions | |
|---|---|---|---|---|---|---|---|
| NMI pulse width | $t_{NMIW}$ | 200 | — | — | ns | NMI digital filter disabled (NMIFLTE.NFLTEN = 0) | $t_{Pcyc} \times 2 \leq 200$ ns |
| | | $t_{Pcyc} \times 2$*1 | — | — | | | $t_{Pcyc} \times 2 > 200$ ns |
| | | 200 | — | — | | NMI digital filter enabled (NMIFLTE.NFLTEN = 1) | $t_{NMICK} \times 3 \leq 200$ ns |
| | | $t_{NMICK} \times 3.5$*2 | — | — | | | $t_{NMICK} \times 3 > 200$ ns |
| IRQ pulse width | $t_{IRQW}$ | 200 | — | — | ns | IRQ digital filter disabled (IRQFLTE0.FLTENi = 0) | $t_{Pcyc} \times 2 \leq 200$ ns |
| | | $t_{Pcyc} \times 2$*1 | — | — | | | $t_{Pcyc} \times 2 > 200$ ns |
| | | 200 | — | — | | IRQ digital filter enabled (IRQFLTE0.FLTENi = 1) | $t_{IRQCK} \times 3 \leq 200$ ns |
| | | $t_{IRQCK} \times 3.5$*3 | — | — | | | $t_{IRQCK} \times 3 > 200$ ns |

Note: • 200 ns minimum in software standby mode.
Note 1. $t_{Pcyc}$ indicates the cycle of PCLKB.
Note 2. $t_{NMICK}$ indicates the cycle of the NMI digital filter sampling clock.
Note 3. $t_{IRQCK}$ indicates the cycle of the IRQi digital filter sampling clock (i = 0 to 7).

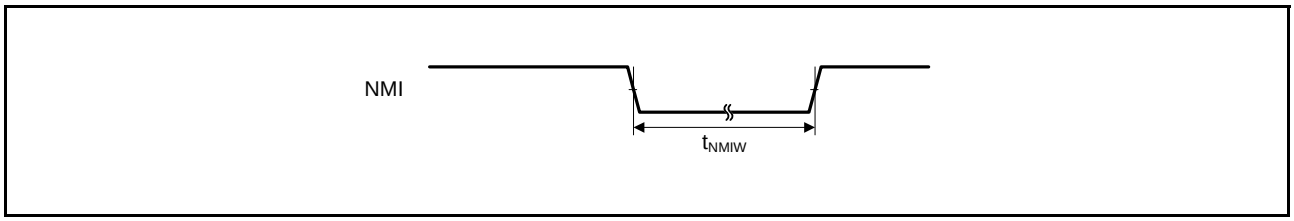



**Figure 36.31 NMI Interrupt Input Timing**

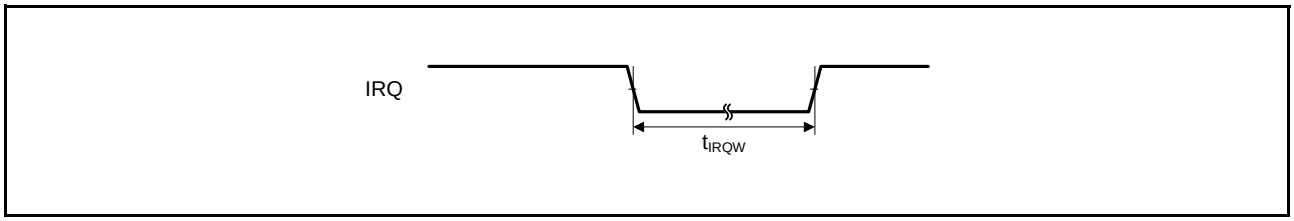



**Figure 36.32 IRQ Interrupt Input Timing**

## 36.3.5 Timing of On-Chip Peripheral Modules

**Table 36.30 Timing of On-Chip Peripheral Modules (1)**

Conditions: VCC = AVCC0 = VCC_USB = 1.8 to 3.6 V, VSS = AVSS0 = VREFL0 = VSS_USB = 0 V, $T_a$ = –40 to +105°C

| Item | | | | Symbol | Min. | Max. | Unit*1 | Test Conditions |
|---|---|---|---|---|---|---|---|---|
| I/O ports | Input data pulse width | | | $t_{PRW}$ | 1.5 | — | $t_{Pcyc}$ | Figure 36.33 |
| MTU2 | Input capture input pulse width | | Single-edge setting | $t_{TICW}$ | 1.5 | — | $t_{Pcyc}$ | Figure 36.34 |
| | | | Both-edge setting | | 2.5 | — | | |
| | Timer clock pulse width | | Single-edge setting | $t_{TCKWH}$, $t_{TCKWL}$ | 1.5 | — | $t_{Pcyc}$ | Figure 36.35 |
| | | | Both-edge setting | | 2.5 | — | | |
| | | | Phase counting mode | | 2.5 | — | | |
| POE | POE# input pulse width | | | $t_{POEW}$ | 1.5 | — | $t_{Pcyc}$ | Figure 36.36 |
| SCI | Input clock cycle | | Asynchronous | $t_{Scyc}$ | 4 | — | $t_{Pcyc}$ | Figure 36.37 |
| | | | Clock synchronous | | 6 | — | | |
| | Input clock pulse width | | | $t_{SCKW}$ | 0.4 | 0.6 | $t_{Scyc}$ | |
| | Input clock rise time | | | $t_{SCKr}$ | — | 20 | ns | |
| | Input clock fall time | | | $t_{SCKf}$ | — | 20 | ns | |
| | Output clock cycle | | Asynchronous | $t_{Scyc}$ | 16 | — | $t_{Pcyc}$ | Figure 36.38 |
| | | | Clock synchronous | | 4 | — | | |
| | Output clock pulse width | | | $t_{SCKW}$ | 0.4 | 0.6 | $t_{Scyc}$ | |
| | Output clock rise time | | | $t_{SCKr}$ | — | 20 | ns | |
| | Output clock fall time | | | $t_{SCKf}$ | — | 20 | ns | |
| | Transmit data delay time (master) | Clock synchronous | | $t_{TXD}$ | — | 40 | ns | |
| | Transmit data delay time (slave) | Clock synchronous | 2.7 V or above | | — | 65 | ns | |
| | | | 1.8 V or above | | — | 100 | ns | |
| | Receive data setup time (master) | Clock synchronous | 2.7 V or above | $t_{RXS}$ | 65 | — | ns | |
| | | | 1.8 V or above | | 90 | — | ns | |
| | Receive data setup time (slave) | Clock synchronous | | | 40 | — | ns | |
| | Receive data hold time | Clock synchronous | | $t_{RXH}$ | 40 | — | ns | |
| A/D converter | Trigger input pulse width | | | $t_{TRGW}$ | 1.5 | — | $t_{Pcyc}$ | Figure 36.39 |
| CAC | CACREF input pulse width | | $t_{Pcyc} \le t_{cac}$*2 | $t_{CACREF}$ | 4.5 $t_{cac}$ + 3 $t_{Pcyc}$ | — | ns | |
| | | | $t_{Pcyc} > t_{cac}$*2 | | 5 $t_{cac}$ + 6.5 $t_{Pcyc}$ | | | |
| CLKOUT | CLKOUT pin output cycle*4 | | VCC = 2.7 V or above | $t_{Ccyc}$ | 125 | — | ns | |
| | | | VCC = 1.8 V or above | | 250 | | | |
| | CLKOUT pin high pulse width*3 | | VCC = 2.7 V or above | $t_{CH}$ | 35 | — | ns | |
| | | | VCC = 1.8 V or above | | 70 | | | |
| | CLKOUT pin low pulse width*3 | | VCC = 2.7 V or above | $t_{CL}$ | 35 | — | ns | |
| | | | VCC = 1.8 V or above | | 70 | | | |
| | CLKOUT pin output rise time | | VCC = 2.7 V or above | $t_{Cr}$ | — | 15 | ns | |
| | | | VCC = 1.8 V or above | | | 30 | | |
| | CLKOUT pin output fall time | | VCC = 2.7 V or above | $t_{Cf}$ | — | 15 | ns | |
| | | | VCC = 1.8 V or above | | | 30 | | |

Note 1. $t_{Pcyc}$: PCLK cycle
Note 2. $t_{cac}$: CAC count clock source cycle
Note 3. When the LOCO is selected as the clock output source (CKOCR.CKOSEL[2:0] bits = 000b), set the clock output division ratio selection to divided by 2 (CKOCR.CKODIV[2:0] bits = 001b).
Note 4. When the EXTAL external clock input or an oscillator is used with divided by 1 (CKOCR.CKOSEL[2:0] bits = 010b and CKOCR.CKODIV[2:0] bits = 000b) to output from CLKOUT, the above should be satisfied with an input duty cycle of 45 to 55%.

### Table 36.31 Timing of On-Chip Peripheral Modules (2)

Conditions: VCC = AVCC0 = VCC_USB = 1.8 to 3.6 V, VSS = AVSS0 = VREFL0 = VSS_USB = 0 V, $T_a$ = –40 to +105°C, C = 30 pF

| Item | | | | Symbol | Min. | Max. | Unit | Test Conditions |
|---|---|---|---|---|---|---|---|---|
| RSPI | RSPCK clock cycle | Master | | $t_{SPcyc}$ | 2 | 4096 | $t_{Pcyc}$ *1 | Figure 36.41 |
| | | Slave | | | 8 | 4096 | | |
| | RSPCK clock high pulse width | Master | | $t_{SPCKWH}$ | $(t_{SPcyc} - t_{SPCKr} - t_{SPCKf})/2 - 3$ | — | ns | |
| | | Slave | | | $(t_{SPcyc} - t_{SPCKr} - t_{SPCKf})/2$ | — | | |
| | RSPCK clock low pulse width | Master | | $t_{SPCKWL}$ | $(t_{SPcyc} - t_{SPCKr} - t_{SPCKf})/2 - 3$ | — | ns | |
| | | Slave | | | $(t_{SPcyc} - t_{SPCKr} - t_{SPCKf})/2$ | — | | |
| | RSPCK clock rise/fall time | Output | 2.7 V or above | $t_{SPCKr}$, $t_{SPCKf}$ | — | 10 | ns | |
| | | | 1.8 V or above | | — | 15 | | |
| | | Input | | | — | 1 | µs | |
| | Data input setup time | Master | 2.7 V or above | $t_{SU}$ | 10 | — | ns | Figure 36.42 to Figure 36.45 |
| | | | 1.8 V or above | | 30 | — | | |
| | | Slave | | | $25 - t_{Pcyc}$ | — | | |
| | Data input hold time | Master | RSPCK set to a division ratio other than PCLKB divided by 2 | $t_H$ | $t_{Pcyc}$ | — | ns | |
| | | | RSPCK set to PCLKB divided by 2 | $t_{HF}$ | 0 | — | | |
| | | Slave | | $t_H$ | $20 + 2 \times t_{Pcyc}$ | — | | |
| | SSL setup time | Master | | $t_{LEAD}$ | $–30 + N^{*2} \times t_{SPcyc}$ | — | ns | |
| | | Slave | | | 2 | — | $t_{Pcyc}$ | |
| | SSL hold time | Master | | $t_{LAG}$ | $–30 + N^{*3} \times t_{SPcyc}$ | — | ns | |
| | | Slave | | | 2 | — | $t_{Pcyc}$ | |
| | Data output delay time | Master | 2.7 V or above | $t_{OD}$ | — | 14 | ns | |
| | | | 1.8 V or above | | — | 30 | | |
| | | Slave | 2.7 V or above | | — | $3 \times t_{Pcyc} + 65$ | | |
| | | | 1.8 V or above | | — | $3 \times t_{Pcyc} + 105$ | | |
| | Data output hold time | Master | 2.7 V or above | $t_{OH}$ | 0 | — | ns | |
| | | | 1.8 V or above | | –20 | — | | |
| | | Slave | | | 0 | — | | |
| | Successive transmission delay time | Master | | $t_{TD}$ | $t_{SPcyc} + 2 \times t_{Pcyc}$ | $8 \times t_{SPcyc} + 2 \times t_{Pcyc}$ | ns | |
| | | Slave | | | $4 \times t_{Pcyc}$ | — | | |
| | MOSI and MISO rise/fall time | Output | 2.7 V or above | $t_{Dr}$, $t_{Df}$ | — | 10 | ns | |
| | | | 1.8 V or above | | — | 20 | | |
| | | Input | | | — | 1 | µs | |
| | SSL rise/fall time | Output | | $t_{SSLr}$, $t_{SSLf}$ | — | 20 | ns | |
| | | Input | | | — | 1 | µs | |
| | Slave access time | | 2.7 V or above | $t_{SA}$ | — | 6 | $t_{Pcyc}$ | Figure 36.44, Figure 36.45 |
| | | | 1.8 V or above | | — | 7 | | |
| | Slave output release time | | 2.7 V or above | $t_{REL}$ | — | 5 | $t_{Pcyc}$ | |
| | | | 1.8 V or above | | — | 6 | | |

Note 1. $t_{Pcyc}$: PCLK cycle
Note 2. N: An integer from 1 to 8 that can be set by the RSPI clock delay register (SPCKD)
Note 3. N: An integer from 1 to 8 that can be set by the RSPI slave select negation delay register (SSLND)

**Table 36.32 Timing of On-Chip Peripheral Modules (3)**

Conditions: VCC = AVCC0 = VCC_USB = 1.8 to 3.6 V, VSS = AVSS0 = VREFL0 = VSS_USB = 0 V, $T_a$ = –40 to +105°C

| Item | | | Symbol | Min. | Max. | Unit*1 | Test Conditions |
|---|---|---|---|---|---|---|---|
| Simple SPI | SCK clock cycle output (master) | | $t_{SPcyc}$ | 4 | 65536 | $t_{Pcyc}$ | Figure 36.41 |
| | SCK clock cycle input (slave) | | | 6 | 65536 | | |
| | SCK input clock high pulse width | | $t_{SPCKWH}$ | 0.4 | 0.6 | $t_{SPcyc}$ | |
| | SCK input clock low pulse width | | $t_{SPCKWL}$ | 0.4 | 0.6 | $t_{SPcyc}$ | |
| | SCK clock rise/fall time | | $t_{SPCKr}$, $t_{SPCKf}$ | — | 20 | ns | |
| | Data input setup time (master) | 2.7 V or above | $t_{SU}$ | 65 | — | ns | Figure 36.42, Figure 36.43 |
| | | 1.8 V or above | | 95 | — | | |
| | Data input setup time (slave) | | | 40 | — | | |
| | Data input hold time | | $t_H$ | 40 | — | ns | |
| | SSL input setup time | | $t_{LEAD}$ | 3 | — | $t_{Pcyc}$ | |
| | SSL input hold time | | $t_{LAG}$ | 3 | — | $t_{Pcyc}$ | |
| | Data output delay time (master) | | $t_{OD}$ | — | 40 | ns | |
| | Data output delay time (slave) | 2.7 V or above | | — | 65 | | |
| | | 1.8 V or above | | — | 85 | | |
| | Data output hold time (master) | 2.7 V or above | $t_{OH}$ | –10 | — | ns | |
| | | 1.8 V or above | | –20 | — | | |
| | Data output hold time (slave) | | | –10 | — | | |
| | Data rise/fall time | | $t_{Dr}$, $t_{Df}$ | — | 20 | ns | |
| | SSL input rise/fall time | | $t_{SSLr}$, $t_{SSLf}$ | — | 20 | ns | |
| | Slave access time | | $t_{SA}$ | — | 6 | $t_{Pcyc}$ | Figure 36.44, Figure 36.45 |
| | Slave output release time | | $t_{REL}$ | — | 6 | $t_{Pcyc}$ | |

Note 1. $t_{Pcyc}$: PCLK cycle

**Table 36.33 Timing of On-Chip Peripheral Modules (4)**

Conditions: VCC = AVCC0 = VCC_USB = 2.7 to 3.6 V, VSS = AVSS0 = VREFL0 = VSS_USB = 0 V, fPCLKB ≤ 32 MHz, $T_a$ = –40 to +105°C

| Item | | Symbol | Min.*1,*2 | Max. | Unit | Test Conditions |
|---|---|---|---|---|---|---|
| RIIC (Standard mode, SMBus) | SCL0 cycle time | $t_{SCL}$ | 6 (12) × $t_{IICcyc}$ + 1300 | — | ns | Figure 36.46 |
| | SCL0 high pulse width | $t_{SCLH}$ | 3 (6) × $t_{IICcyc}$ + 300 | — | ns | |
| | SCL0 low pulse width | $t_{SCLL}$ | 3 (6) × $t_{IICcyc}$ + 300 | — | ns | |
| | SCL0, SDA0 rise time | $t_{Sr}$ | — | 1000 | ns | |
| | SCL0, SDA0 fall time | $t_{Sf}$ | — | 300 | ns | |
| | SCL0, SDA0 spike pulse removal time | $t_{SP}$ | 0 | 1 (4) × $t_{IICcyc}$ | ns | |
| | SDA0 bus free time | $t_{BUF}$ | 3 (6) × $t_{IICcyc}$ + 300 | — | ns | |
| | START condition hold time | $t_{STAH}$ | $t_{IICcyc}$ + 300 | — | ns | |
| | Repeated START condition setup time | $t_{STAS}$ | 1000 | — | ns | |
| | STOP condition setup time | $t_{STOS}$ | 1000 | — | ns | |
| | Data setup time | $t_{SDAS}$ | $t_{IICcyc}$ + 50 | — | ns | |
| | Data hold time | $t_{SDAH}$ | 0 | — | ns | |
| | SCL0, SDA0 capacitive load | $C_b$ | — | 400 | pF | |
| RIIC (Fast mode) | SCL0 cycle time | $t_{SCL}$ | 6 (12) × $t_{IICcyc}$ + 600 | — | ns | Figure 36.46 |
| | SCL0 high pulse width | $t_{SCLH}$ | 3 (6) × $t_{IICcyc}$ + 300 | — | ns | |
| | SCL0 low pulse width | $t_{SCLL}$ | 3 (6) × $t_{IICcyc}$ + 300 | — | ns | |
| | SCL0, SDA0 rise time | $t_{Sr}$ | —*3 | 300 | ns | |
| | SCL0, SDA0 fall time | $t_{Sf}$ | —*3 | 300 | ns | |
| | SCL0, SDA0 spike pulse removal time | $t_{SP}$ | 0 | 1 (4) × $t_{IICcyc}$ | ns | |
| | SDA0 bus free time | $t_{BUF}$ | 3 (6) × $t_{IICcyc}$ + 300 | — | ns | |
| | START condition hold time | $t_{STAH}$ | $t_{IICcyc}$ + 300 | — | ns | |
| | Repeated START condition setup time | $t_{STAS}$ | 300 | — | ns | |
| | STOP condition setup time | $t_{STOS}$ | 300 | — | ns | |
| | Data setup time | $t_{SDAS}$ | $t_{IICcyc}$ + 50 | — | ns | |
| | Data hold time | $t_{SDAH}$ | 0 | — | ns | |
| | SCL0, SDA0 capacitive load | $C_b$ | — | 400 | pF | |

Note: • $t_{IICcyc}$: RIIC internal reference count clock (IICφ) cycle

Note 1. The value in parentheses is used when the ICMR3.NF[1:0] bits are set to 11b while a digital filter is enabled with the ICFER.NFE bit = 1.

Note 2. $C_b$ indicates the total capacity of the bus line.

Note 3. The minimum tsr and tsf specifications for fast mode are not set.

**Table 36.34 Timing of On-Chip Peripheral Modules (5)**

Conditions: VCC = AVCC0 = VCC_USB = 2.7 to 3.6 V, VSS = AVSS0 = VREFL0 = VSS_USB = 0 V, fPCLKB ≤ 32 MHz,
$T_a$ = –40 to +105°C

| Item | | Symbol | Min.*1 | Max. | Unit | Test Conditions |
|---|---|---|---|---|---|---|
| Simple IIC (Standard mode) | SDA0 rise time | $t_{Sr}$ | — | 1000 | ns | Figure 36.46 |
| | SDA0 fall time | $t_{Sf}$ | — | 300 | ns | |
| | SDA0 spike pulse removal time | $t_{SP}$ | 0 | 4 × $t_{pcyc}$*2 | ns | |
| | Data setup time | $t_{SDAS}$ | 250 | — | ns | |
| | Data hold time | $t_{SDAH}$ | 0 | — | ns | |
| | SCL0, SDA0 capacitive load | $C_b$ | — | 400 | pF | |
| Simple IIC (Fast mode) | SCL0, SDA0 rise time | $t_{Sr}$ | — | 300 | ns | Figure 36.46 |
| | SCL0, SDA0 fall time | $t_{Sf}$ | — | 300 | ns | |
| | SCL0, SDA0 spike pulse removal time | $t_{SP}$ | 0 | 4 × $t_{pcyc}$*2 | ns | |
| | Data setup time | $t_{SDAS}$ | 100 | — | ns | |
| | Data hold time | $t_{SDAH}$ | 0 | — | ns | |
| | SCL0, SDA0 capacitive load | $C_b$ | — | 400 | pF | |

Note: • $t_{Pcyc}$: PCLK cycle
Note 1. $C_b$ indicates the total capacity of the bus line.
Note 2. This applies when the SMR.CKS[1:0] bits = 00b and the SNFR.NFCS[2:0] bits = 010b while the SNFR.NFE bit = 1 and the digital filter is enabled.

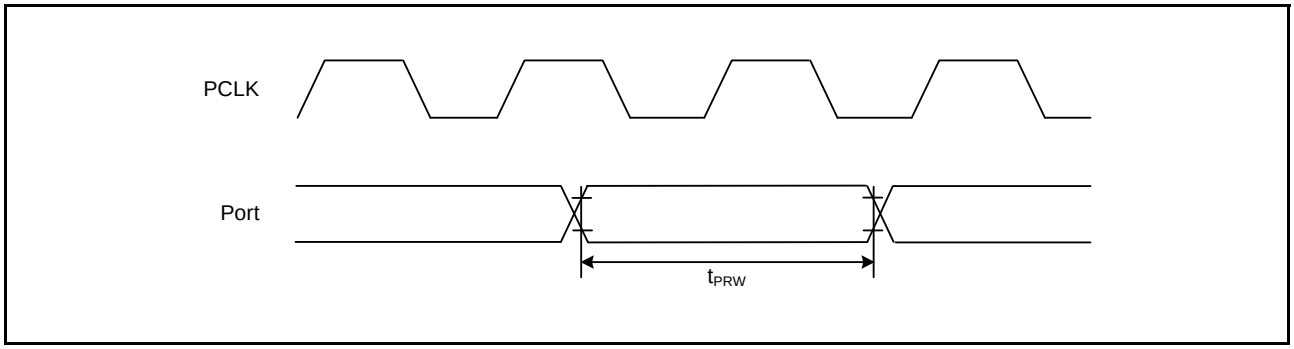



**Figure 36.33 I/O Port Input Timing**

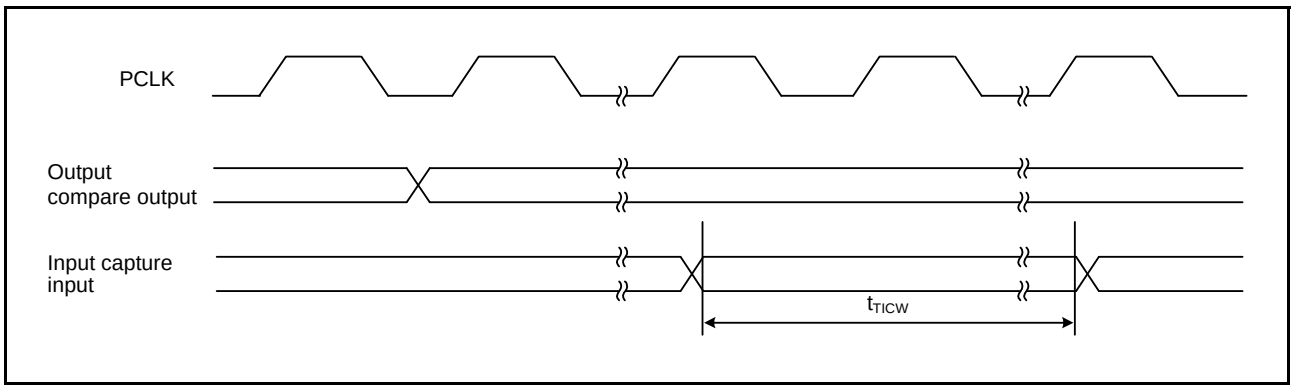



**Figure 36.34 MTU2 Input/Output Timing**

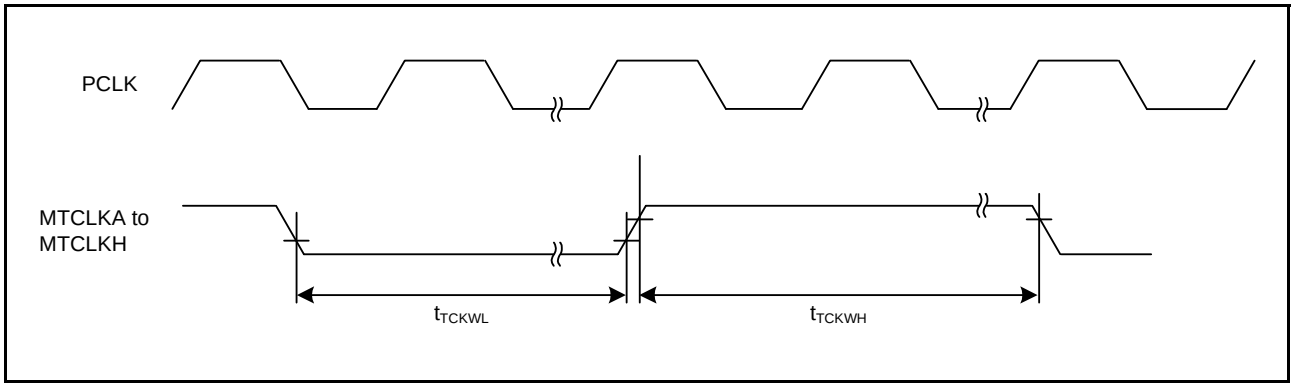



**Figure 36.35 MTU2 Clock Input Timing**

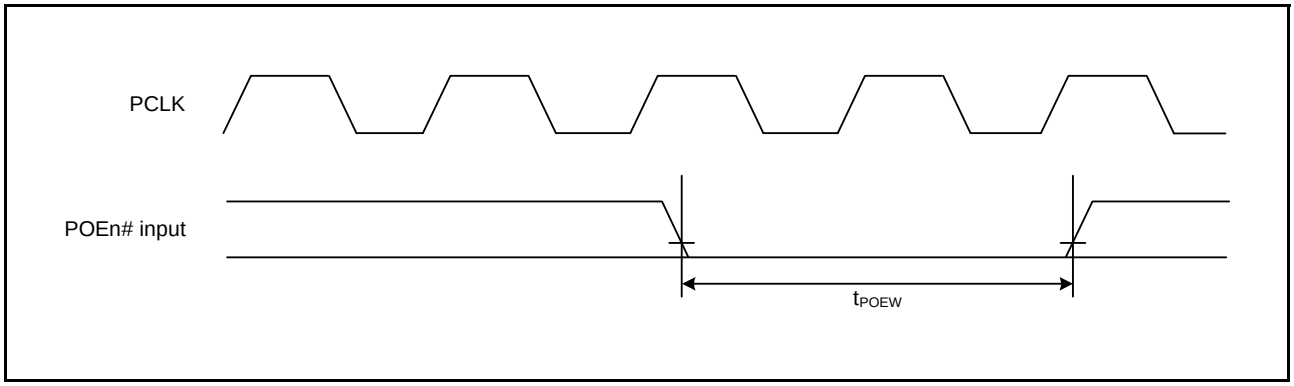



**Figure 36.36 POE# Input Timing**

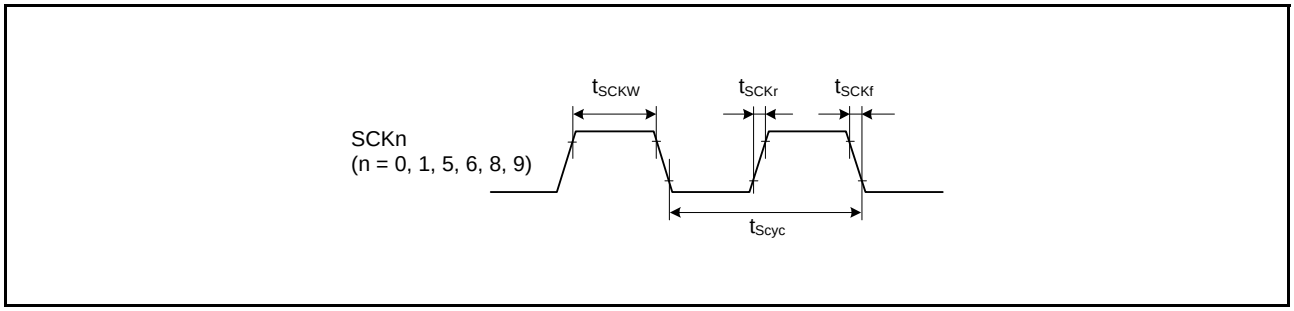



**Figure 36.37 SCK Clock Input Timing**

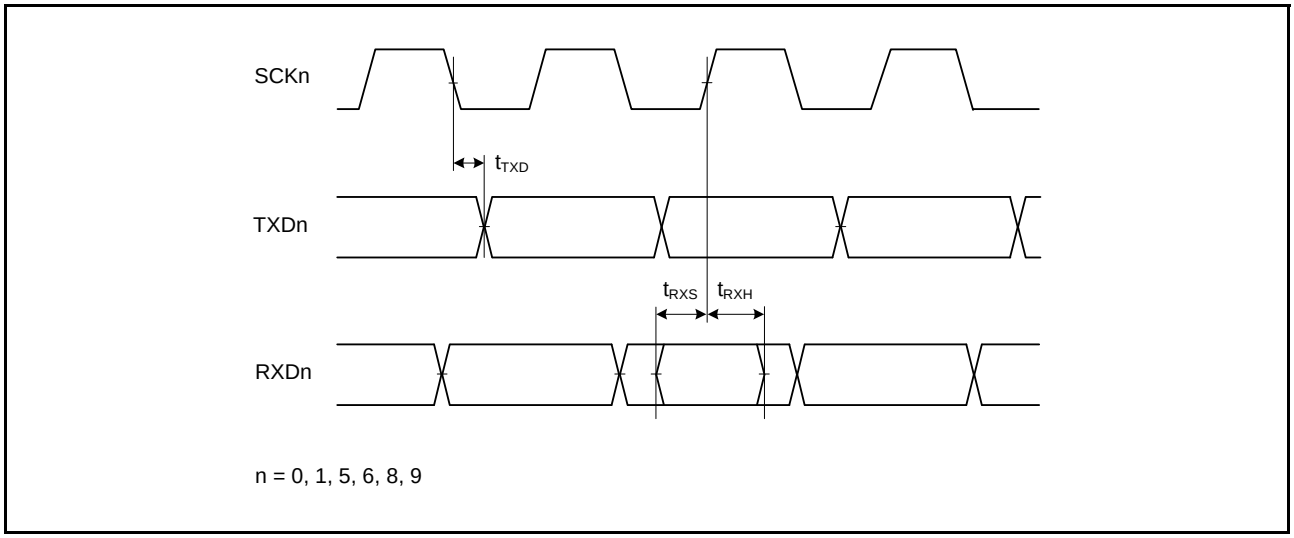



**Figure 36.38 SCI Input/Output Timing: Clock Synchronous Mode**

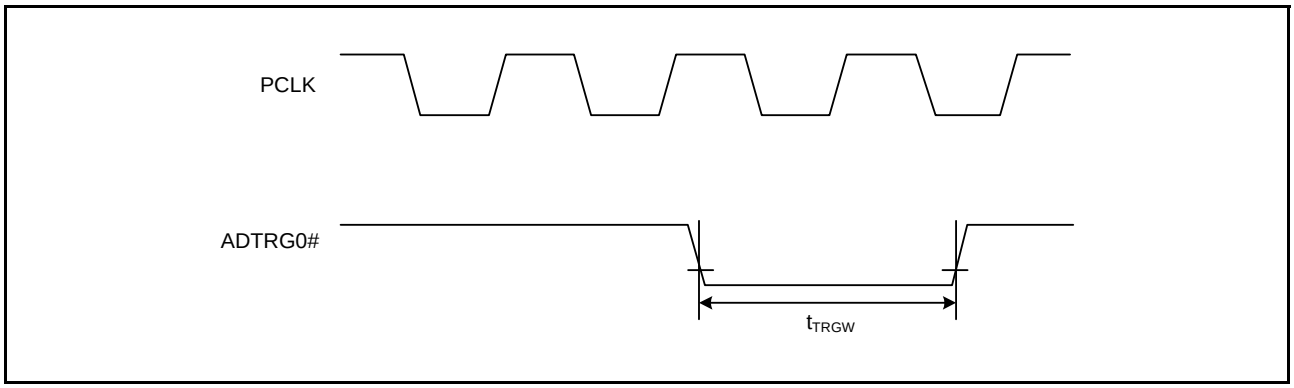



**Figure 36.39 A/D Converter External Trigger Input Timing**

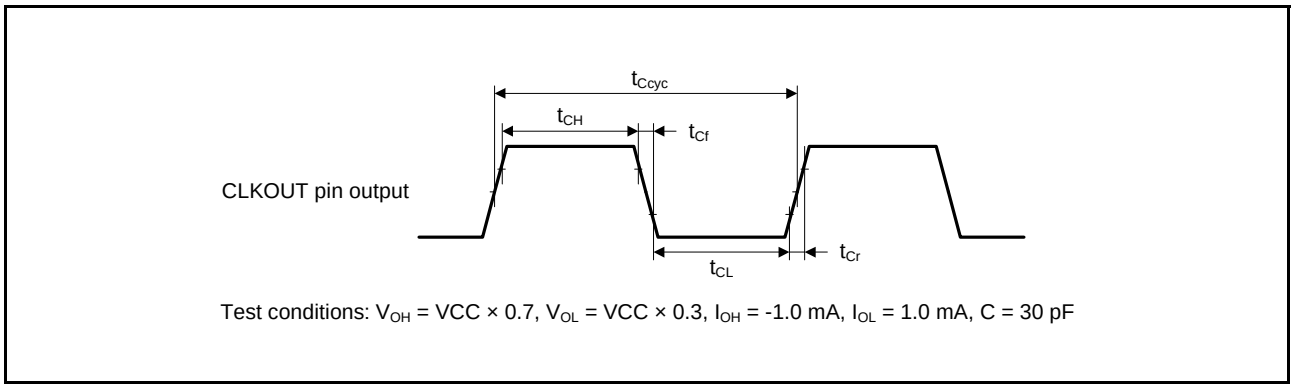



**Figure 36.40 CLKOUT Output Timing**

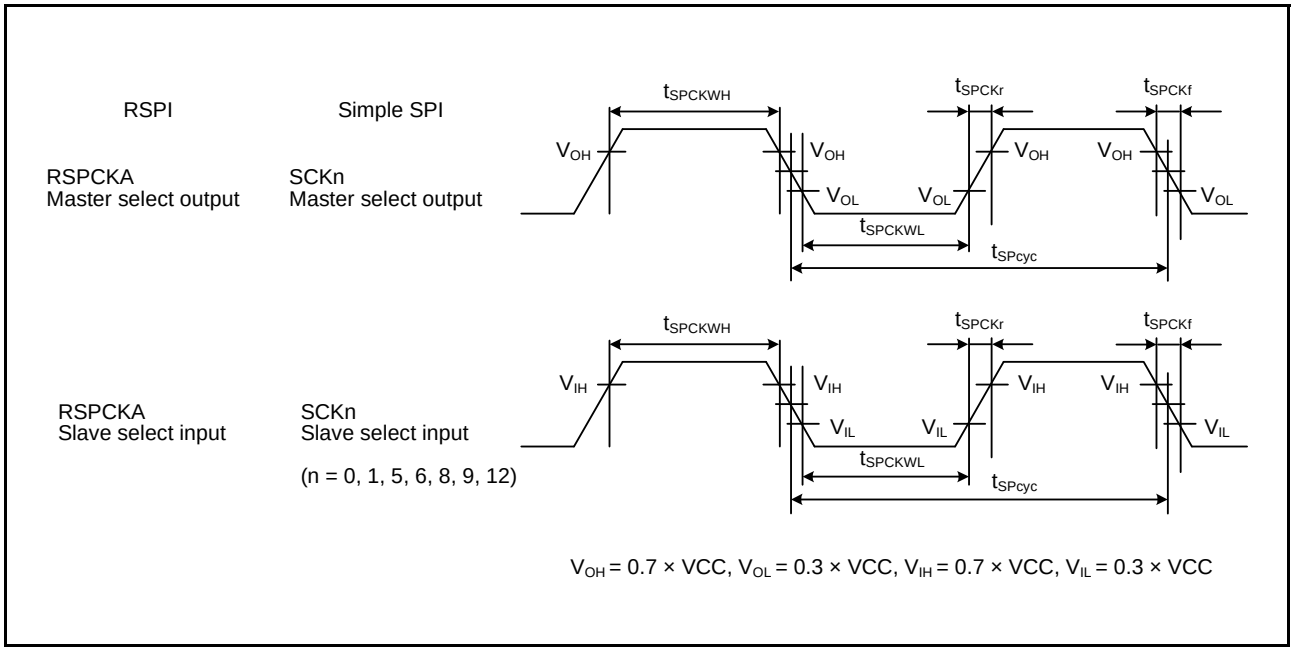



**Figure 36.41 RSPI Clock Timing and Simple SPI Clock Timing**

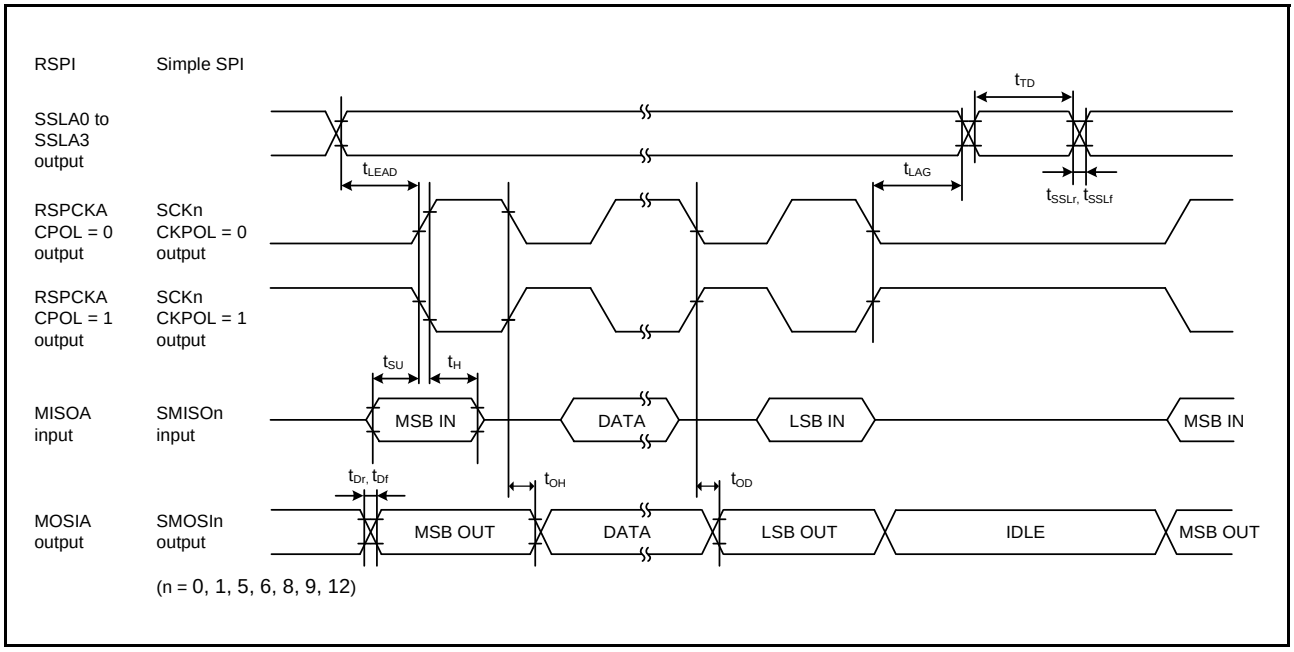



**Figure 36.42 RSPI Timing (Master, CPHA = 0) and Simple SPI Timing (Master, CKPH = 1)**

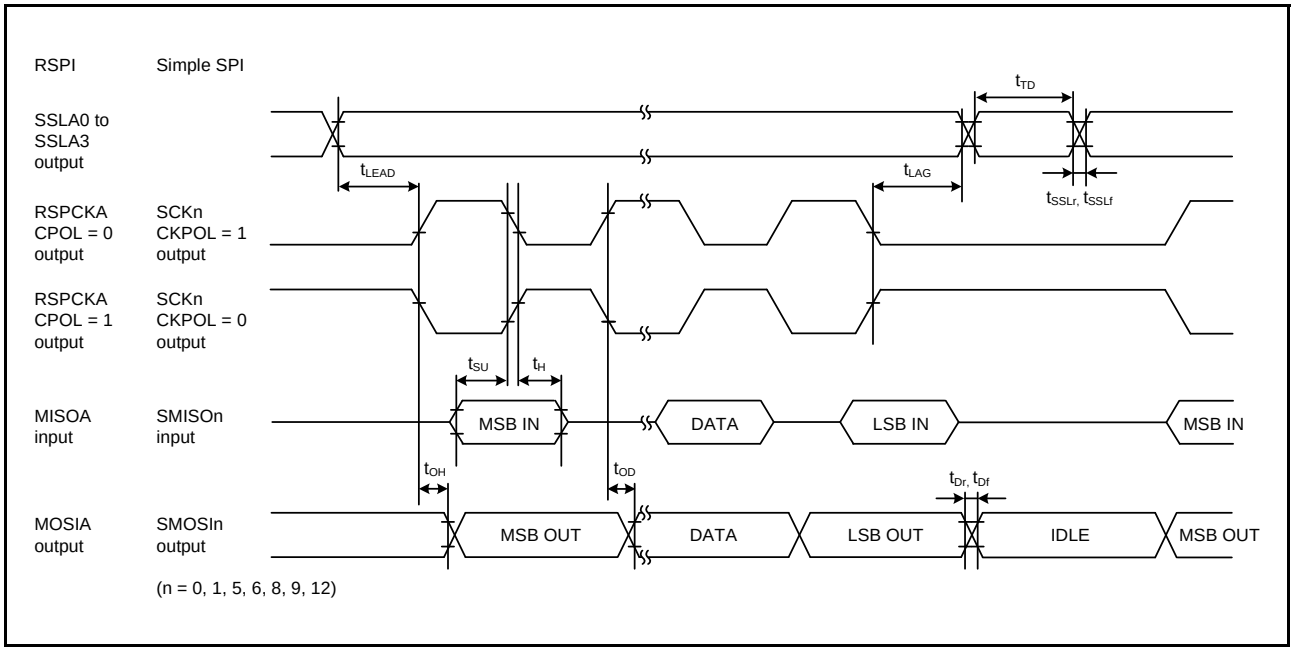



**Figure 36.43 RSPI Timing (Master, CPHA = 1) and Simple SPI Timing (Master, CKPH = 0)**

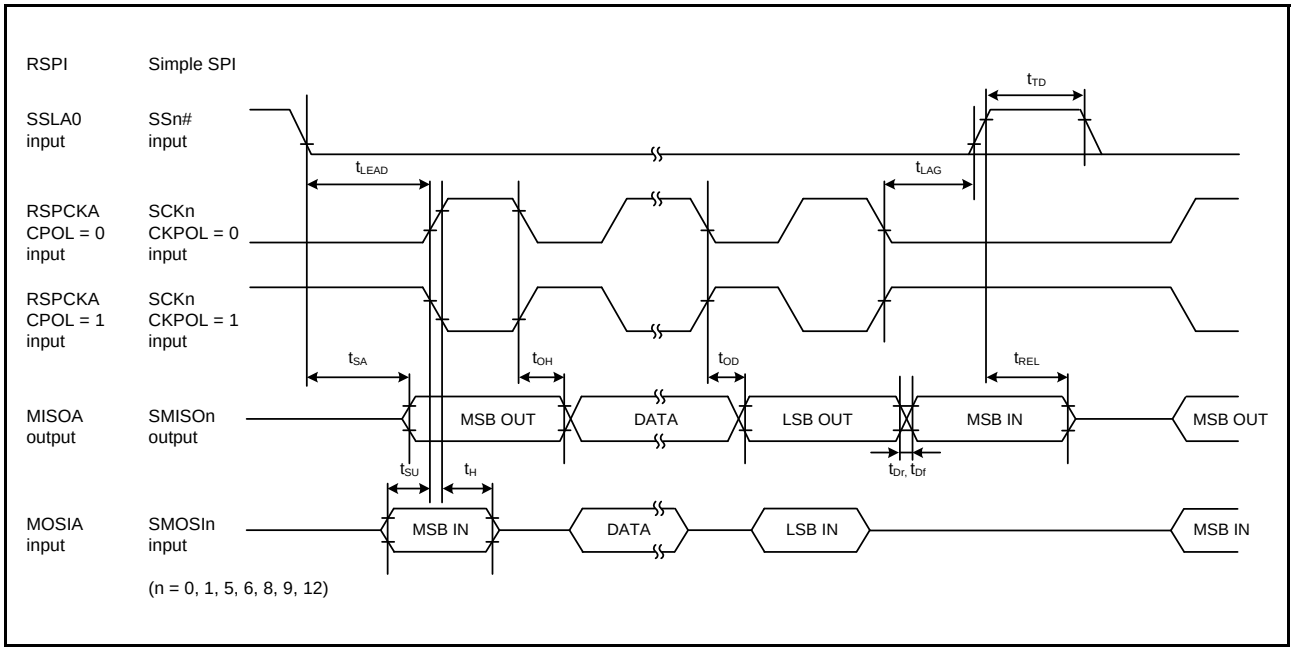



**Figure 36.44 RSPI Timing (Slave, CPHA = 0) and Simple SPI Timing (Slave, CKPH = 1)**

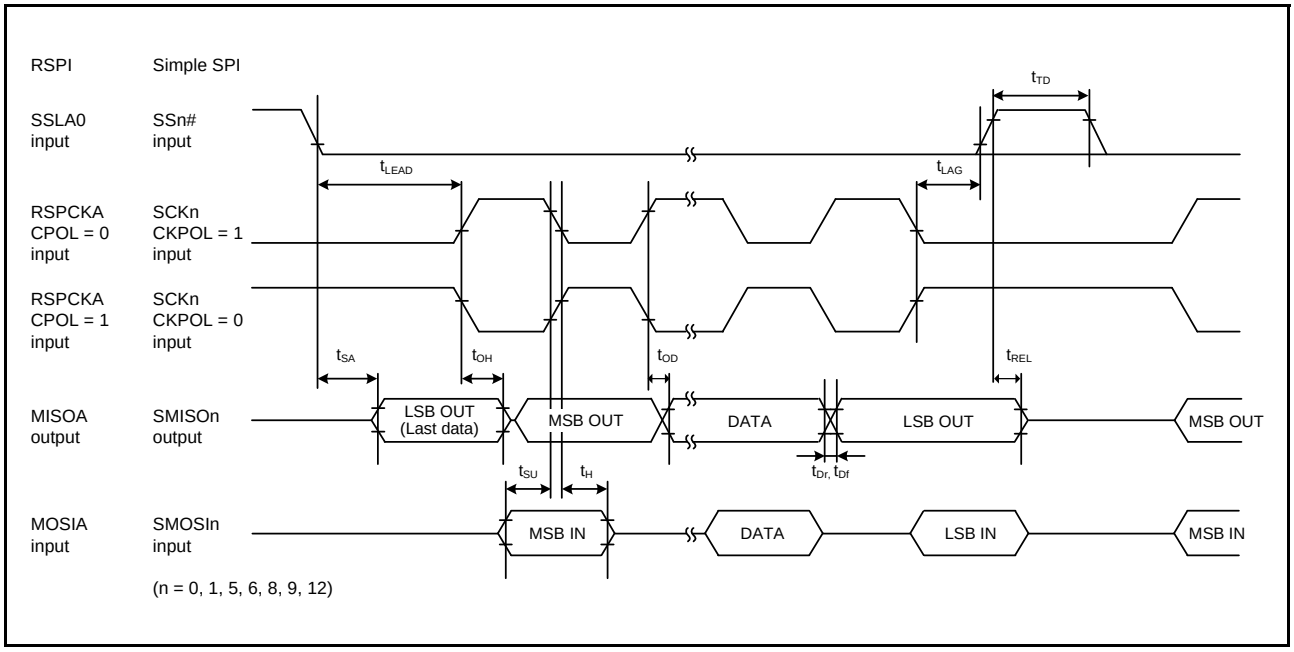



**Figure 36.45 RSPI Timing (Slave, CPHA = 1) and Simple SPI Timing (Slave, CKPH = 0)**

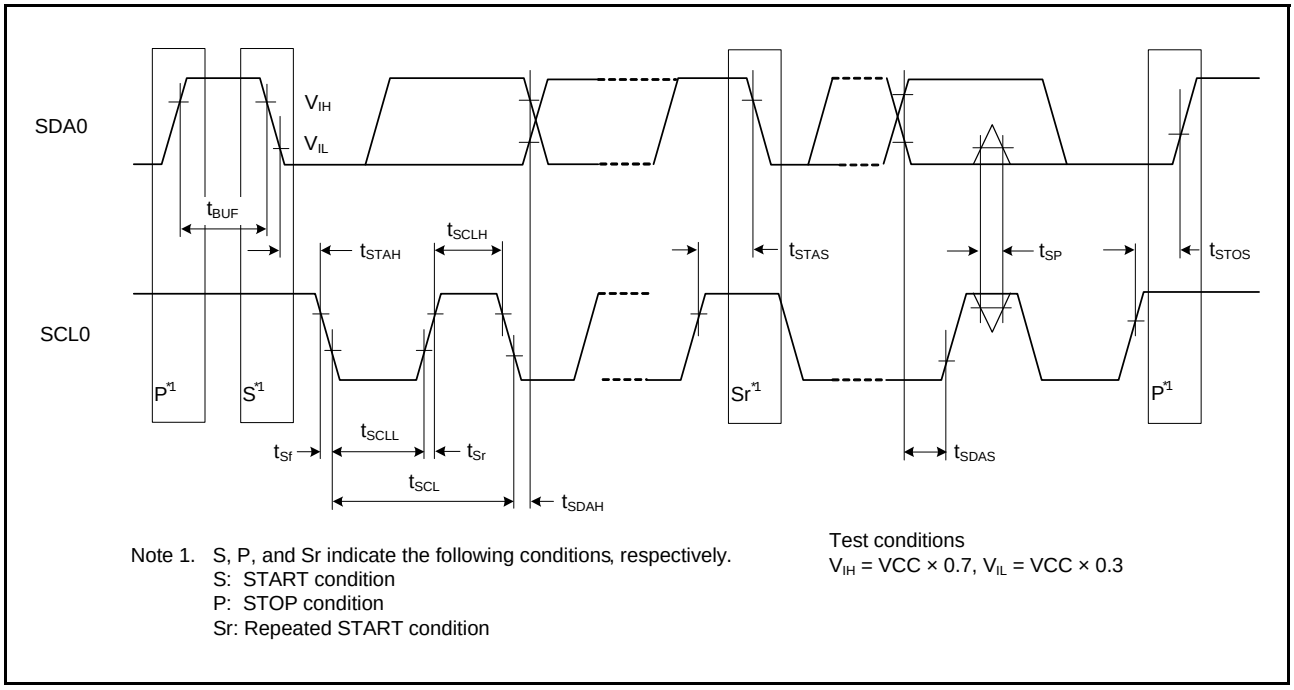



**Figure 36.46 RIIC Bus Interface Input/Output Timing and Simple IIC Bus Interface Input/Output Timing**

## 36.4 USB Characteristics

**Table 36.35 USB Characteristics (DP and DM Pin Characteristics)**

Conditions: VCC = AVCC0 = VCC_USB = 3.0 to 3.6 V, VSS = AVSS0 = VREFL0 = VSS_USB = 0 V, $T_a$ = –40 to +105°C

| Item | | | Symbol | Min. | Max. | Unit | Test Conditions | |
|---|---|---|---|---|---|---|---|---|
| Input characteristics | Input high level voltage | | $V_{IH}$ | 2.0 | — | V | | |
| | Input low level voltage | | $V_{IL}$ | — | 0.8 | V | | |
| | Differential input sensitivity | | $V_{DI}$ | 0.2 | — | V | \| DP – DM \| | |
| | Differential common mode range | | $V_{CM}$ | 0.8 | 2.5 | V | | |
| Output characteristics | Output high level voltage | | $V_{OH}$ | 2.8 | VCC_USB | V | $I_{OH}$ = –200 µA | |
| | Output low level voltage | | $V_{OL}$ | 0.0 | 0.3 | V | $I_{OL}$= 2 mA | |
| | Cross-over voltage | | $V_{CRS}$ | 1.3 | 2.0 | V | | Figure 36.47 Figure 36.48 |
| | Rise time | FS | $t_r$ | 4 | 20 | ns | | |
| | | LS | | 75 | 300 | | | |
| | Fall time | FS | $t_f$ | 4 | 20 | ns | | |
| | | LS | | 75 | 300 | | | |
| | Rise/fall time ratio | FS | $t_r/t_f$ | 90 | 111.11 | % | $t_r/t_f$ | |
| | | LS | | 80 | 125 | | | |
| | Output resistance | | $Z_{DRV}$ | 28 | 44 | Ω | (Adjusting the resistance of external elements is not necessary.) | |
| VBUS characteristics | VBUS input voltage | | $V_{IH}$ | VCC × 0.8 | — | V | | |
| | | | $V_{IL}$ | — | VCC × 0.2 | V | | |
| | VBUS (P16) input leakage current | | \| $I_{VBUSIN}$ \| | — | 10 | µA | USB0_VBUS = 5.5V | |
| Pull-up, pull-down | Pull-down resistor | | $R_{PD}$ | 14.25 | 24.80 | kΩ | | |
| | Pull-up resistor | | $R_{PUI}$ | 0.9 | 1.575 | kΩ | During idle state | |
| | | | $R_{PUA}$ | 1.425 | 3.09 | kΩ | During reception | |
| Battery Charging Specification Ver 1.2 | D+ sink current | | $I_{DP\_SINK}$ | 25 | 175 | µA | | |
| | D- sink current | | $I_{DM\_SINK}$ | 25 | 175 | µA | | |
| | DCD source current | | $I_{DP\_SRC}$ | 7 | 13 | µA | | |
| | Data detection voltage | | $V_{DAT\_REF}$ | 0.25 | 0.4 | V | | |
| | D+ source current | | $V_{DP\_SRC}$ | 0.5 | 0.7 | V | Output current = 250 µA | |
| | D- source current | | $V_{DM\_SRC}$ | 0.5 | 0.7 | V | Output current = 250 µA | |

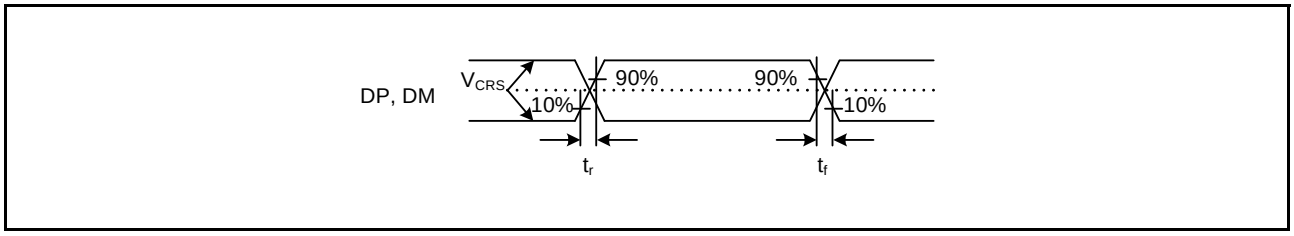



**Figure 36.47 DP and DM Output Timing**

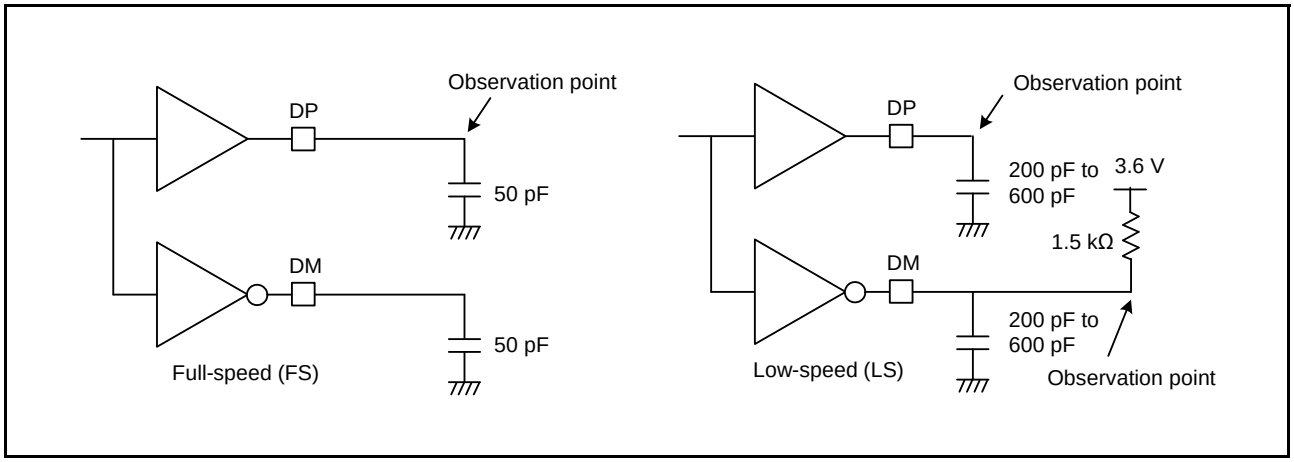



**Figure 36.48 Test Circuit**

## 36.5 A/D Conversion Characteristics

**Table 36.36 A/D Conversion Characteristics (1)**

Conditions: VCC = AVCC0 = VREFH0 = 2.7 to 3.6 V, VSS = AVSS0 = VREFL0 = VSS_USB = 0 V, PCLKD = 4 to 32 MHz, $T_a$ = –40 to +105°C

| Item | | Min. | Typ. | Max. | Unit | Test Conditions |
|---|---|---|---|---|---|---|
| Resolution | | — | — | 12 | Bit | |
| Conversion time*1 (Operation at PCLKD = 32 MHz) | Permissible signal source impedance (Max.) = 0.3 kΩ | 1.031 (0.313)*2 | — | — | µs | High-precision channel<br>ADCSR.ADHSC bit = 1<br>ADSSTRn.SST[7:0] bits = 09h |
| | | 1.375 (0.641)*2 | — | — | | Normal-precision channel<br>ADCSR.ADHSC bit = 1<br>ADSSTRn.SST[7:0] bits = 14h |
| Offset error | | — | ±0.5 | ±4.5 | LSB | High-precision channel<br>PJ6PFS.ASEL bit = 1<br>PJ7PFS.ASEL bit = 1 |
| | | | | ±6.0 | LSB | Other than above |
| Full-scale error | | — | ±0.75 | ±4.5 | LSB | High-precision channel<br>PJ6PFS.ASEL bit = 1<br>PJ7PFS.ASEL bit = 1 |
| | | | | ±6.0 | LSB | Other than above |
| Quantization error | | — | ±0.5 | — | LSB | |
| Absolute accuracy | | — | ±1.25 | ±5.0 | LSB | High-precision channel<br>PJ6PFS.ASEL bit = 1<br>PJ7PFS.ASEL bit = 1 |
| | | | | ±8.0 | LSB | Other than above |
| DNL differential nonlinearity error | | — | ±1.0 | — | LSB | |
| INL integral nonlinearity error | | — | ±1.0 | ±3.0 | LSB | |

Note: • The characteristics apply when no pin functions other than A/D converter input are used. Absolute accuracy includes quantization errors. Offset error, full-scale error, DNL differential nonlinearity error, and INL integral nonlinearity error do not include quantization errors.

Note 1. The conversion time is the sum of the sampling time and the comparison time. As the test conditions, the number of sampling states is indicated.

Note 2. The value in parentheses indicates the sampling time.

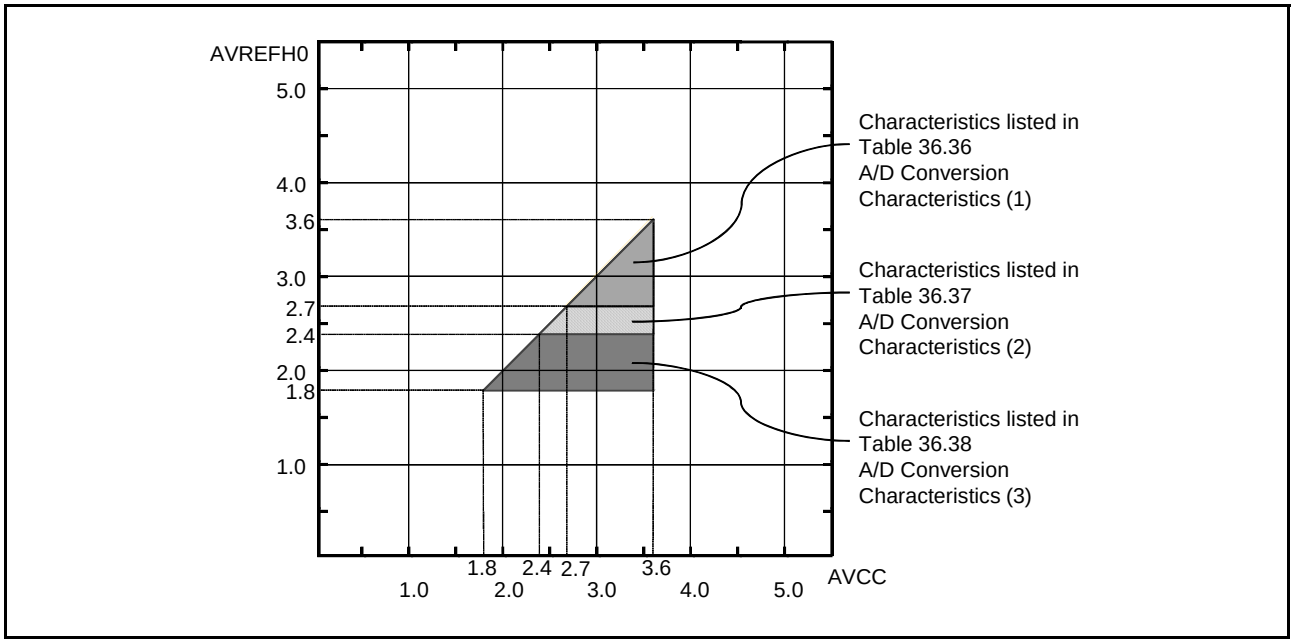



**Figure 36.49 AVCC to AVREFH Voltage Range**

### Table 36.37 A/D Conversion Characteristics (2)

Conditions: VCC = AVCC0 = VREFH0 = 2.4 to 2.7 V, VSS = AVSS0 = VREFL0 = VSS_USB = 0 V, PCLKD = 4 to 16 MHz,
$T_a$ = –40 to +105°C

| Item | | Min. | Typ. | Max. | Unit | Test Conditions |
|---|---|---|---|---|---|---|
| Resolution | | — | — | 12 | Bit | |
| Conversion time*1 (Operation at fPCLKD = 16 MHz) | Permissible signal source impedance (Max.) = 1.0 kΩ | 2.062 (0.625)*2 | — | — | µs | High-precision channel ADCSR.ADHSC bit = 1 ADSSTRn.SST[7:0] bits = 09h |
| | | 2.750 (1.313)*2 | — | — | µs | Normal-precision channel ADCSR.ADHSC bit = 1 ADSSTRn.SST[7:0] bits = 14h |
| Offset error | | — | ±0.5 | ±6.0 | LSB | |
| Full-scale error | | — | ±1.25 | ±6.0 | LSB | |
| Quantization error | | — | ±0.5 | — | LSB | |
| Absolute accuracy | | — | ±3.0 | ±8.0 | LSB | |
| DNL differential nonlinearity error | | — | ±1.0 | — | LSB | |
| INL integral nonlinearity error | | — | ±1.5 | ±3.0 | LSB | |

Note: • The characteristics apply when no pin functions other than A/D converter input are used. Absolute accuracy includes quantization errors. Offset error, full-scale error, DNL differential nonlinearity error, and INL integral nonlinearity error do not include quantization errors.

Note 1. The conversion time is the sum of the sampling time and the comparison time. As the test conditions, the number of sampling states is indicated.

Note 2. The value in parentheses indicates the sampling time.

### Table 36.38 A/D Conversion Characteristics (3)

Conditions: VCC = AVCC0 = VREFH0 = 1.8 to 2.4 V, VSS = AVSS0 = VREFL0 = VSS_USB = 0 V, PCLKD = 1 to 8 MHz,
$T_a$ = –40 to +105°C

| Item | | Min. | Typ. | Max. | Unit | Test Conditions |
|---|---|---|---|---|---|---|
| Resolution | | — | — | 12 | Bit | |
| Conversion time*1 (Operation at PCLKD = 8 MHz) | Permissible signal source impedance (Max.) = 5.0 kΩ | 4.875 (1.250)*2 | — | — | µs | High-precision channel ADCSR.ADHSC bit = 0 ADSSTRn.SST[7:0] bits = 09h |
| | | 6.250 (2.625)*2 | — | — | | Normal-precision channel ADCSR.ADHSC bit = 0 ADSSTRn.SST[7:0] bits = 14h |
| Offset error | | — | ±0.5 | ±24.0 | LSB | |
| Full-scale error | | — | ±1.25 | ±24.0 | LSB | |
| Quantization error | | — | ±0.5 | — | LSB | |
| Absolute accuracy | | — | ±2.75 | ±32.0 | LSB | |
| DNL differential nonlinearity error | | — | ±1.0 | — | LSB | |
| INL integral nonlinearity error | | — | ±1.25 | ±12.0 | LSB | |

Note: • The characteristics apply when no pin functions other than A/D converter input are used. Absolute accuracy includes quantization errors. Offset error, full-scale error, DNL differential nonlinearity error, and INL integral nonlinearity error do not include quantization errors.

Note 1. The conversion time is the sum of the sampling time and the comparison time. As the test conditions, the number of sampling states is indicated.

Note 2. The value in parentheses indicates the sampling time.

**Table 36.39 A/D Converter Channel Classification**

| Classification | Channel | Conditions | Remarks |
|---|---|---|---|
| High-precision channel | AN000 to AN004, AN006 | AVCC0 = 1.8 to 3.6 V | Pins AN000 to AN004 and AN006 cannot be used as digital outputs when the A/D converter is in use. |
| Normal-precision channel | AN008 to AN015 | | |
| Internal reference voltage input channel | Internal reference voltage | AVCC0 = 2.0 to 3.6 V | |
| Temperature sensor input channel | Temperature sensor output | AVCC0 = 2.0 to 3.6 V | |

**Table 36.40 A/D Internal Reference Voltage Characteristics**

Conditions: VCC = AVCC0 = VCC_USB = 2.0 to 3.6 V*1, VSS = AVSS0 = VREFL0 = VSS_USB = 0 V, $T_a$ = –40 to +105°C

| Item | Min. | Typ. | Max. | Unit | Test Conditions |
|---|---|---|---|---|---|
| Internal reference voltage input channel*2 | 1.36 | 1.43 | 1.50 | V | |

Note 1. The internal reference voltage cannot be selected for input channels when AVCC0 < 2.0 V.
Note 2. The A/D internal reference voltage indicates the voltage when the internal reference voltage is input to the A/D converter.

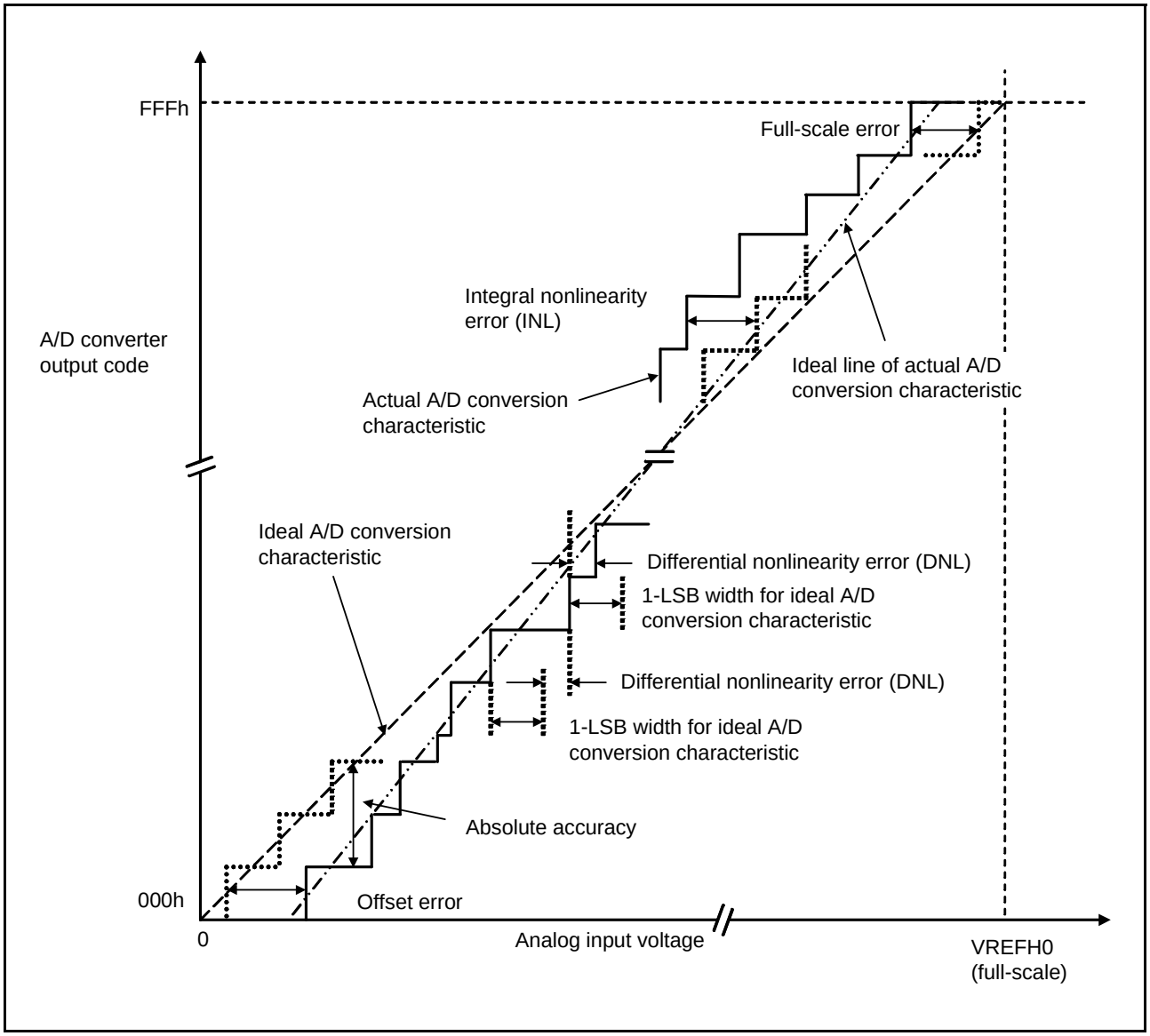



**Figure 36.50 Illustration of A/D Converter Characteristic Terms**

### Absolute accuracy

Absolute accuracy is the difference between output code based on the theoretical A/D conversion characteristics, and the actual A/D conversion result. When measuring absolute accuracy, the voltage at the midpoint of the width of analog input voltage (1-LSB width), that can meet the expectation of outputting an equal code based on the theoretical A/D conversion characteristics, is used as an analog input voltage. For example, if 12-bit resolution is used and if reference voltage (VREFH0 = 3.072 V), then 1-LSB width becomes 0.75 mV, and 0 mV, 0.75 mV, 1.5 mV, ... are used as analog input voltages.

If analog input voltage is 6 mV, absolute accuracy = ±5 LSB means that the actual A/D conversion result is in the range of 003h to 00Dh though an output code, 008h, can be expected from the theoretical A/D conversion characteristics.

### Integral nonlinearity error (INL)

Integral nonlinearity error is the maximum deviation between the ideal line when the measured offset and full-scale errors are zeroed, and the actual output code.

### Differential nonlinearity error (DNL)

Differential nonlinearity error is the difference between 1-LSB width based on the ideal A/D conversion characteristics and the width of the actually output code.

### Offset error

Offset error is the difference between a transition point of the ideal first output code and the actual first output code.

### Full-scale error

Full-scale error is the difference between a transition point of the ideal last output code and the actual last output code.

## 36.6 D/A Conversion Characteristics

**Table 36.41 D/A Conversion Characteristics (1)**

Conditions: VCC = AVCC0 = VCC_USB = 1.8 to 3.6 V, VSS = AVSS0 = VREFL0 = VSS_USB = 0 V, fPCLKB ≤ 32 MHz, $T_a$ = –40 to +105°C

| Item | | Min. | Typ. | Max. | Unit | Test Conditions |
|---|---|---|---|---|---|---|
| Resolution | | — | — | 8 | Bit | |
| Conversion time | VCC = 2.7 to 3.6 V | — | — | 3.0 | µs | 35-pF capacitive load |
| | VCC = 1.6 to 2.7 V | — | — | 6.0 | | |
| Absolute accuracy | VCC = 2.4 to 3.6 V | — | — | ±3.0 | LSB | 2-MΩ resistive load |
| | VCC = 1.8 to 2.4 V | — | — | ±3.5 | | |
| | VCC = 2.4 to 3.6 V | — | — | ±2.0 | LSB | 4-MΩ resistive load |
| | VCC = 1.8 to 2.4 V | — | — | ±2.5 | | |
| RO output resistance | | — | 6.4 | — | kΩ | |

## 36.7 Temperature Sensor Characteristics

**Table 36.42 Temperature Sensor Characteristics**

Conditions: VCC = AVCC0 = VCC_USB = 2.0 to 3.6 V, VSS = AVSS0 = VREFL0 = VSS_USB = 0 V, $T_a$ = –40 to +105°C

| Item | Symbol | Min. | Typ. | Max. | Unit | Test Conditions |
|---|---|---|---|---|---|---|
| Relative accuracy | — | — | ±1.5 | — | °C | 2.4 V or above |
| | | — | ±2.0 | — | | Below 2.4 V |
| Temperature slope | — | — | –3.65 | — | mV/°C | |
| Output voltage (at 25°C) | — | — | 1.05 | — | V | VCC = 3.3 V |
| Temperature sensor start time | $t_{START}$ | — | — | 5 | µs | |
| Sampling time | — | 5 | — | — | µs | |

## 36.8 Power-On Reset Circuit and Voltage Detection Circuit Characteristics

**Table 36.43 Power-On Reset Circuit and Voltage Detection Circuit Characteristics (1)**

Conditions: VCC = AVCC0 = VCC_USB, VSS = AVSS0 = VREFL0 = VSS_USB = 0 V, $T_a$ = –40 to +105°C

| Item | | Symbol | Min. | Typ. | Max. | Unit | Test Conditions |
|---|---|---|---|---|---|---|---|
| Voltage detection level | Power-on reset (POR) | $V_{POR}$ | 1.35 | 1.50 | 1.65 | V | Figure 36.51, Figure 36.52 |
| | Voltage detection circuit (LVD1)*1 | $V_{det1\_4}$ | 3.00 | 3.10 | 3.20 | V | Figure 36.53 At falling edge VCC |
| | | $V_{det1\_5}$ | 2.91 | 3.00 | 3.09 | | |
| | | $V_{det1\_6}$ | 2.81 | 2.90 | 2.99 | | |
| | | $V_{det1\_7}$ | 2.70 | 2.79 | 2.88 | | |
| | | $V_{det1\_8}$ | 2.60 | 2.68 | 2.76 | | |
| | | $V_{det1\_9}$ | 2.50 | 2.58 | 2.66 | | |
| | | $V_{det1\_A}$ | 2.40 | 2.48 | 2.56 | | |
| | | $V_{det1\_B}$ | 1.99 | 2.06 | 2.13 | | |
| | | $V_{det1\_C}$ | 1.90 | 1.96 | 2.02 | | |
| | | $V_{det1\_D}$ | 1.80 | 1.86 | 1.92 | | |

Note: • These characteristics apply when noise is not superimposed on the power supply. When a setting is made so that the voltage detection level overlaps with that of the voltage detection circuit (LVD2), it cannot be specified which of LVD1 and LVD2 is used for voltage detection.

Note 1. # in the symbol Vdet1_# denotes the value of the LVDLVLR.LVD1LVL[3:0] bits.

**Table 36.44 Power-On Reset Circuit and Voltage Detection Circuit Characteristics (2)**

Conditions: VCC = AVCC0 = VCC_USB, VSS = AVSS0 = VREFL0 = VSS_USB = 0 V, $T_a$ = –40 to +105°C

| Item | | Symbol | Min. | Typ. | Max. | Unit | Test Conditions |
|---|---|---|---|---|---|---|---|
| Voltage detection level | Voltage detection circuit (LVD2)*1 | $V_{det2\_0}$ | 2.71 | 2.90 | 3.09 | V | Figure 36.54 At falling edge VCC |
| | | $V_{det2\_1}$ | 2.43 | 2.60 | 2.77 | | |
| | | $V_{det2\_2}$ | 1.87 | 2.00 | 2.13 | | |
| | | $V_{det2\_3}$*2 | 1.69 | 1.80 | 1.91 | | |
| Wait time after power-on reset cancellation | At normal startup*3 | $t_{POR}$ | — | 9.1 | — | ms | Figure 36.52 |
| | During fast startup time*4 | $t_{POR}$ | — | 1.6 | — | | |
| Wait time after voltage monitoring 1 reset cancellation | Power-on voltage monitoring 1 reset disabled*3 | $t_{LVD1}$ | — | 568 | — | µs | Figure 36.53 |
| | Power-on voltage monitoring 1 reset enabled*4 | | — | 100 | — | | |
| Wait time after voltage monitoring 2 reset cancellation | | $t_{LVD2}$ | — | 100 | — | µs | Figure 36.54 |
| Response delay time | | $t_{det}$ | — | — | 350 | µs | Figure 36.51 |
| Minimum VCC down time*5 | | $t_{VOFF}$ | 350 | — | — | µs | Figure 36.51, VCC = 1.0 V or above |
| Power-on reset enable time | | $t_{W(POR)}$ | 1 | — | — | ms | Figure 36.52, VCC = below 1.0 V |
| LVD operation stabilization time (after LVD is enabled) | | $Td_{(E-A)}$ | — | — | 300 | µs | Figure 36.53, Figure 36.54 |
| Hysteresis width (LVD1 and LVD2) | | $V_{LVH}$ | — | 70 | — | mV | Vdet1_4 selected |
| | | | — | 60 | — | | Vdet1_5 to 9, LVD2 selected |
| | | | — | 50 | — | | When selection is from among Vdet1_A to B. |
| | | | — | 40 | — | | When selection is from among Vdet1_C to D. |

Note: • These characteristics apply when noise is not superimposed on the power supply. When a setting is made so that the voltage detection level overlaps with that of the voltage detection circuit (LVD1), it cannot be specified which of LVD1 and LVD2 is used for voltage detection.

Note 1. # in the symbol Vdet2_# denotes the value of the LVDLVLR.LVD2LVL[3:0] bits.

Note 2. Vdet2_3 selection can be used only when the CMPA2 pin input voltage is selected and cannot be used when the power supply voltage (VCC) is selected.

Note 3. When OFS1.(STUPLVD1REN, FASTSTUP) = 11b.

Note 4. When OFS1.(STUPLVD1REN, FASTSTUP) ≠ 11b.

Note 5. The minimum VCC down time indicates the time when VCC is below the minimum value of voltage detection levels $V_{POR}$, $V_{det0}$, $V_{det1}$, and $V_{det2}$ for the POR/LVD.

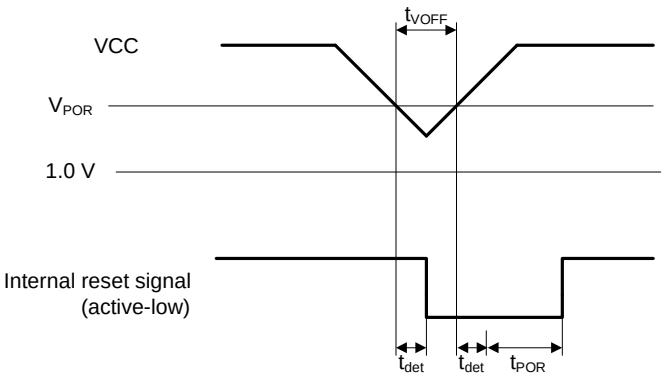
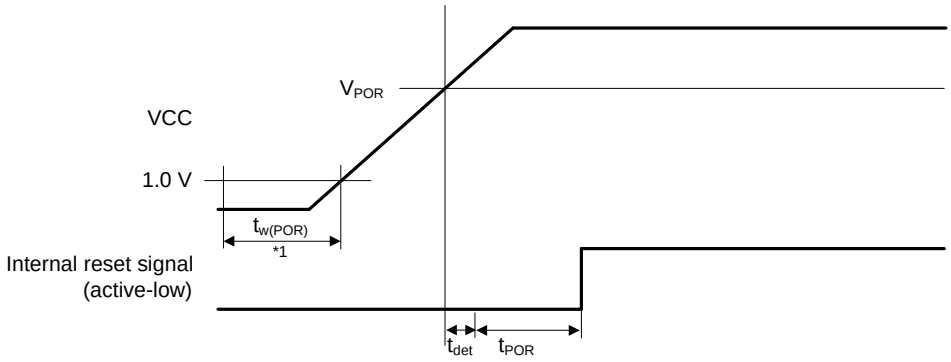



**Figure 36.51 Voltage Detection Reset Timing**




Note 1. $t_{w(por)}$ is the time required for a power-on reset to be enabled while the external power VCC is being held below the valid voltage (1.0 V).
When VCC turns on, maintain $t_{w(por)}$ for 1.0 ms or more.

**Figure 36.52 Power-On Reset Timing**

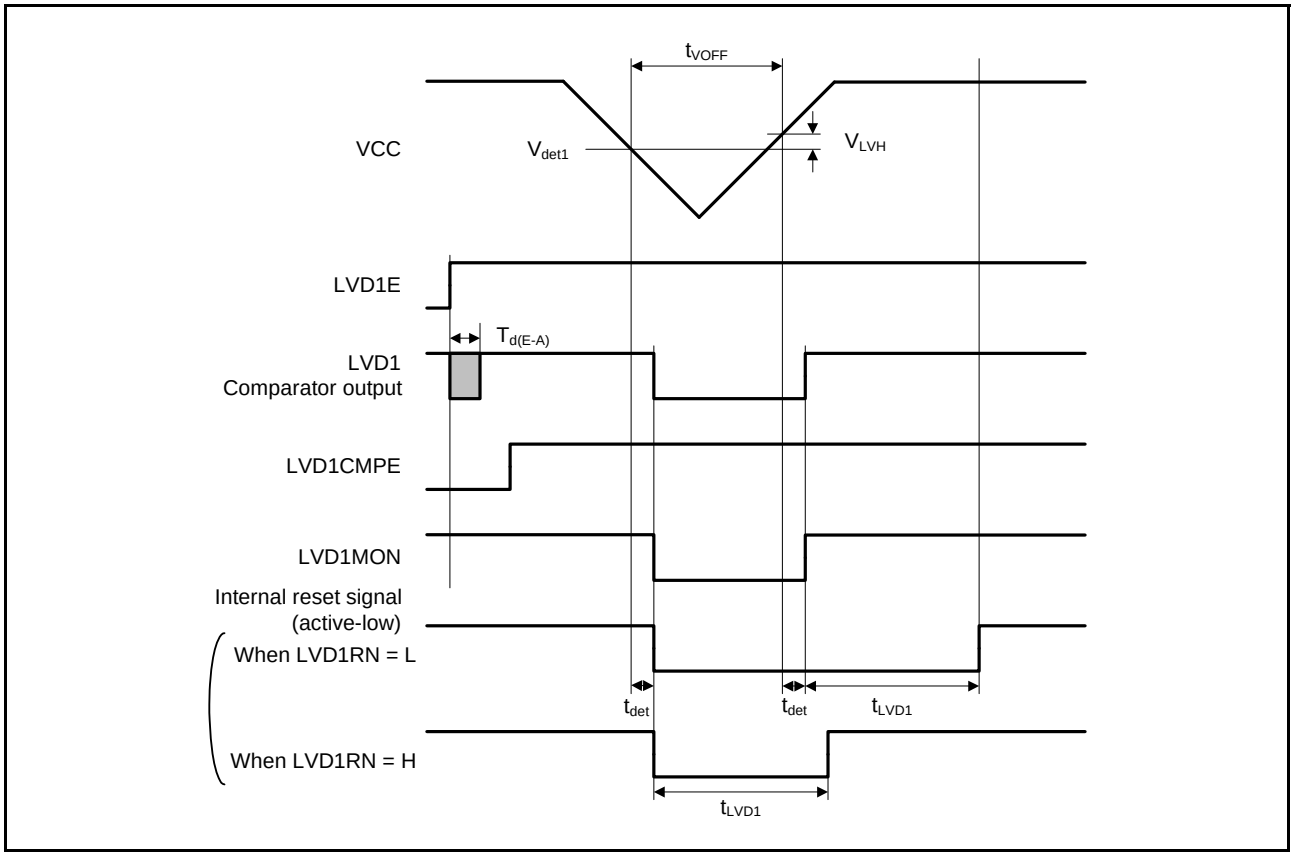



**Figure 36.53 Voltage Detection Circuit Timing ($V_{det1}$)**

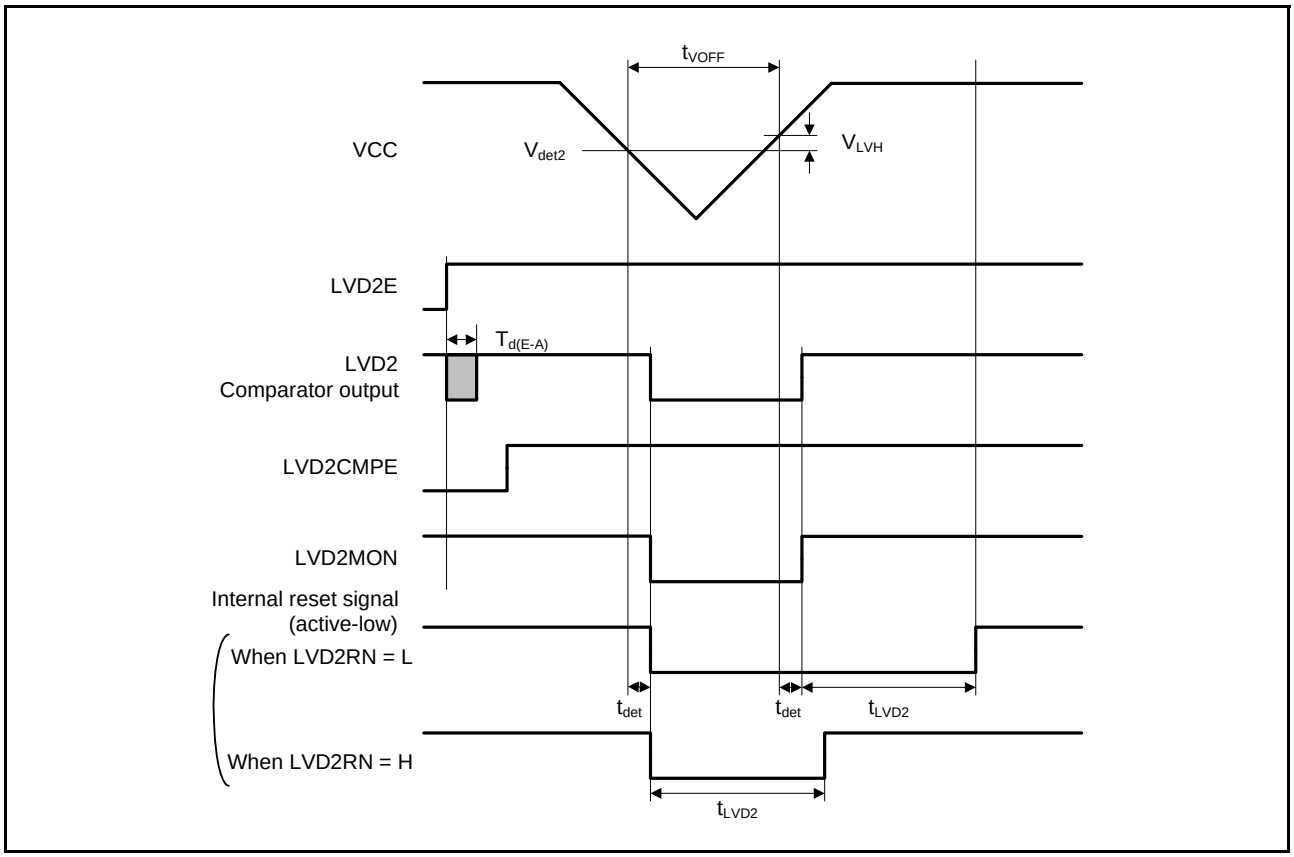



**Figure 36.54 Voltage Detection Circuit Timing ($V_{det2}$)**

## 36.9 Oscillation Stop Detection Timing

**Table 36.45 Oscillation Stop Detection Circuit Characteristics**

Conditions: VCC = AVCC0 = VCC_USB = 1.8 to 3.6 V, VSS = AVSS0 = VREFL0 = VSS_USB = 0 V, $T_a$ = –40 to +105°C

| Item | Symbol | Min. | Typ. | Max. | Unit | Test Conditions |
|---|---|---|---|---|---|---|
| Detection time | $t_{dr}$ | — | — | 1 | ms | Figure 36.55 |

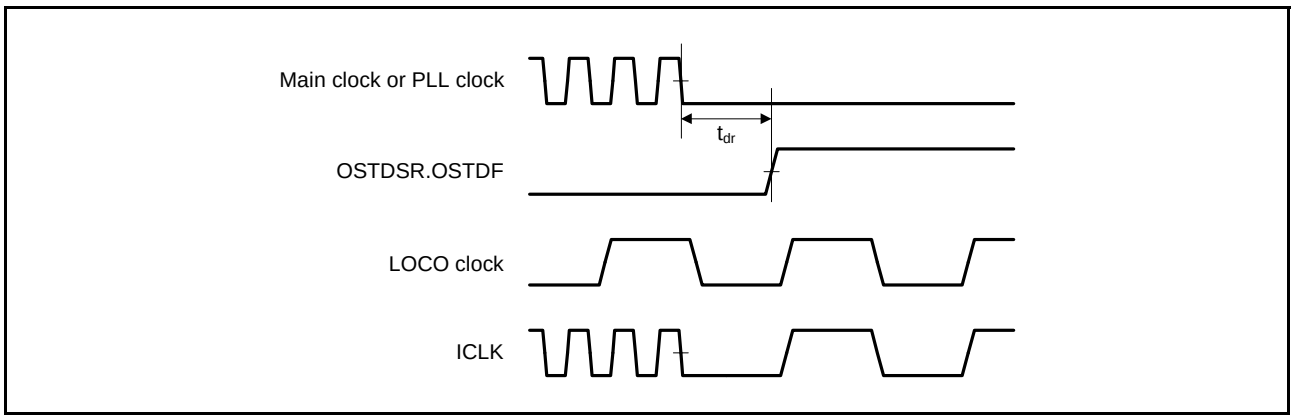



**Figure 36.55 Oscillation Stop Detection Timing**

## 36.10 ROM (Flash Memory for Code Storage) Characteristics

### Table 36.46 ROM (Flash Memory for Code Storage) Characteristics (1)

| Item | | Symbol | Min. | Typ. | Max. | Unit | Conditions |
|---|---|---|---|---|---|---|---|
| Reprogramming/erasure cycle*1 | | $N_{PEC}$ | 1000 | — | — | Times | |
| Data hold time | After 1000 times of $N_{PEC}$ | $t_{DRP}$ | 20*2, *3 | — | — | Year | Ta = +85°C |

Note 1. Definition of reprogram/erase cycle: The reprogram/erase cycle is the number of erasing for each block. When the reprogram/erase cycle is n times (n = 1000), erasing can be performed n times for each block. For instance, when 4-byte programming is performed 256 times for different addresses in 1-Kbyte block and then the entire block is erased, the reprogram/erase cycle is counted as one. However, programming the same address for several times as one erasing is not enabled (overwriting is prohibited).
Note 2. Characteristic when using the flash memory programmer and the self-programming library provided from Renesas Electronics.
Note 3. This result is obtained from reliability testing.

### Table 36.47 ROM (Flash Memory for Code Storage) Characteristics (2) : high-speed operating mode, middle-speed operating mode

Conditions: VCC = AVCC0 = 2.7 to 3.6 V, VSS = AVSS0 = VREFL0 = 0 V
Temperature range for the programming/erasure operation: $T_a$ = –40 to +105°C

| Item | | Symbol | FCLK = 1 MHz | | | FCLK = 32 MHz | | | Unit |
|---|---|---|---|---|---|---|---|---|---|
| | | | Min. | Typ. | Max. | Min. | Typ. | Max. | |
| Programming time | 4-byte | $t_{P4}$ | — | 1650 | 4910 | — | 100 | 761 | µs |
| Erasure time | 1-Kbyte | $t_{E1K}$ | — | 9.77 | 329 | — | 5.53 | 258 | ms |
| Blank check time | 4-byte | $t_{BC4}$ | — | — | 5000 | — | — | 316 | µs |
| | 1-Kbyte | $t_{BC1K}$ | — | — | 1280 | — | — | 80.7 | ms |

Note: • Does not include the time until each operation of the flash memory is started after instructions are executed by software.
Note: • The lower-limit frequency of FCLK is 1 MHz during programming or erasing of the flash memory. When using FCLK at below 4 MHz, the frequency can be set to 1 MHz, 2 MHz, or 3 MHz. A non-integer frequency such as 1.5 MHz cannot be set.
Note: • The frequency accuracy of FCLK should be ±3.5%. Confirm the frequency accuracy of the clock source.

### Table 36.48 ROM (Flash Memory for Code Storage) Characteristics (3) : middle-speed operating mode

Conditions: VCC = AVCC0 = 1.8 to 3.6 V, VSS = AVSS0 = VREFL0 = 0 V
Temperature range for the programming/erasure operation: $T_a$ = –40 to +105°C

| Item | | Symbol | FCLK = 1 MHz | | | FCLK = 8 MHz | | | Unit |
|---|---|---|---|---|---|---|---|---|---|
| | | | Min. | Typ. | Max. | Min. | Typ. | Max. | |
| Programming time | 4-byte | $t_{P4}$ | — | 1690 | 5380 | — | 290 | 1680 | µs |
| Erasure time | 1-Kbyte | $t_{E1K}$ | — | 9.84 | 331 | — | 6.04 | 275 | ms |
| Blank check time | 4-byte | $t_{BC4}$ | — | — | 4980 | — | — | 973 | µs |
| | 1-Kbyte | $t_{BC1K}$ | — | — | 1270 | — | — | 250 | ms |

Note: • Does not include the time until each operation of the flash memory is started after instructions are executed by software.
Note: • The lower-limit frequency of FCLK is 1 MHz during programming or erasing of the flash memory. When using FCLK at below 4 MHz, the frequency can be set to 1 MHz, 2 MHz, or 3 MHz. A non-integer frequency such as 1.5 MHz cannot be set.
Note: • The frequency accuracy of FCLK should be ±3.5%. Confirm the frequency accuracy of the clock source.

## 36.11 E2 DataFlash Characteristics

### Table 36.49 E2 DataFlash Characteristics (1)

| Item | | Symbol | Min. | Typ. | Max. | Unit | Conditions |
|---|---|---|---|---|---|---|---|
| Reprogramming/erasure cycle*1 | | $N_{DPEC}$ | 100000 | 1000000 | — | Times | |
| Data hold time | After 10000 times of $N_{DPEC}$ | $t_{DDRP}$ | 20*2, *3 | — | — | Year | Ta = +85°C |
| | After 100000 times of $N_{DPEC}$ | | 5*2, *3 | — | — | Year | |
| | After 1000000 times of $N_{DPEC}$ | | — | 1*2, *3 | — | Year | Ta = +25°C |

Note 1. The reprogram/erase cycle is the number of erasing for each block. When the reprogram/erase cycle is n times (n = 100000), erasing can be performed n times for each block. For instance, when 1-byte programming is performed 1000 times for different addresses in 1-byte block and then the entire block is erased, the reprogram/erase cycle is counted as one. However, programming the same address for several times as one erasing is not enabled (overwriting is prohibited).
Note 2. Characteristics when using the flash memory programmer and the self-programming library provided from Renesas Electronics.
Note 3. These results are obtained from reliability testing.

### Table 36.50 E2 DataFlash Characteristics (2) : high-speed operating mode, middle-speed operating mode

Conditions: VCC = AVCC0 = 2.7 to 3.6 V, VSS = AVSS0 = VREFL0 = 0 V

Temperature range for the programming/erasure operation: $T_a$ = –40 to +105°C

| Item | | Symbol | FCLK = 4 MHz | | | FCLK = 32 MHz | | | Unit |
|---|---|---|---|---|---|---|---|---|---|
| | | | Min. | Typ. | Max. | Min. | Typ. | Max. | |
| Programming time | 1-byte | $t_{DP1}$ | — | 1450 | 1740 | — | 83.2 | 484 | µs |
| Erasure time | 1-Kbyte | $t_{DE1K}$ | — | 18.8 | 547 | — | 8.19 | 274 | ms |
| Blank check time | 1-byte | $t_{DBC1}$ | — | — | 1130 | — | — | 70.5 | µs |
| | 1-Kbyte | $t_{DBC1K}$ | — | — | 7590 | — | — | 580 | ms |
| Suspended time during erasing | | $t_{DSED}$ | — | — | 21.5 | — | — | 12.8 | µs |
| DataFlash STOP recovery time | | $t_{DSTOP}$ | 5 | — | — | 5 | — | — | µs |

Note: • Does not include the time until each operation of the flash memory is started after instructions are executed by software.
Note: • The lower-limit frequency of FCLK is 1 MHz during programming or erasing of the flash memory. When using FCLK at below 4 MHz, the frequency can be set to 1 MHz, 2 MHz, or 3 MHz. A non-integer frequency such as 1.5 MHz cannot be set.
Note: • The frequency accuracy of FCLK should be ±3.5%. Confirm the frequency accuracy of the clock source.

### Table 36.51 E2 DataFlash Characteristics (3) : middle-speed operating mode

Conditions: VCC = AVCC0 = 1.8 to 3.6 V, VSS = AVSS0 = VREFL0 = 0 V

Temperature range for the programming/erasure operation: $T_a$ = –40 to +105°C

| Item | | Symbol | FCLK = 4 MHz | | | FCLK = 32 MHz | | | Unit |
|---|---|---|---|---|---|---|---|---|---|
| | | | Min. | Typ. | Max. | Min. | Typ. | Max. | |
| Programming time | 1-byte | $t_{DP1}$ | — | 1490 | 2220 | — | 255 | 1130 | µs |
| Erasure time | 1-Kbyte | $t_{DE1K}$ | — | 18.9 | 498 | — | 7.93 | 280 | ms |
| Blank check time | 1-byte | $t_{DBC1}$ | — | — | 1170 | — | — | 212 | µs |
| | 1-Kbyte | $t_{DBC1K}$ | — | — | 7.39 | — | — | 1.97 | ms |
| Suspended time during erasing | | $t_{DSED}$ | — | — | 33.5 | — | — | 25.5 | µs |
| DataFlash STOP recovery time | | $t_{DSTOP}$ | 720 | — | — | 720 | — | — | ns |

Note: • Does not include the time until each operation of the flash memory is started after instructions are executed by software.
Note: • The lower-limit frequency of FCLK is 1 MHz during programming or erasing of the flash memory. When using FCLK at below 4 MHz, the frequency can be set to 1 MHz, 2 MHz, or 3 MHz. A non-integer frequency such as 1.5 MHz cannot be set.
Note: • The frequency accuracy of FCLK should be ±3.5%. Confirm the frequency accuracy of the clock source.

# Appendix 1. Port States in Each Processing Mode

**Table 1.1 Port States in Each Processing State**

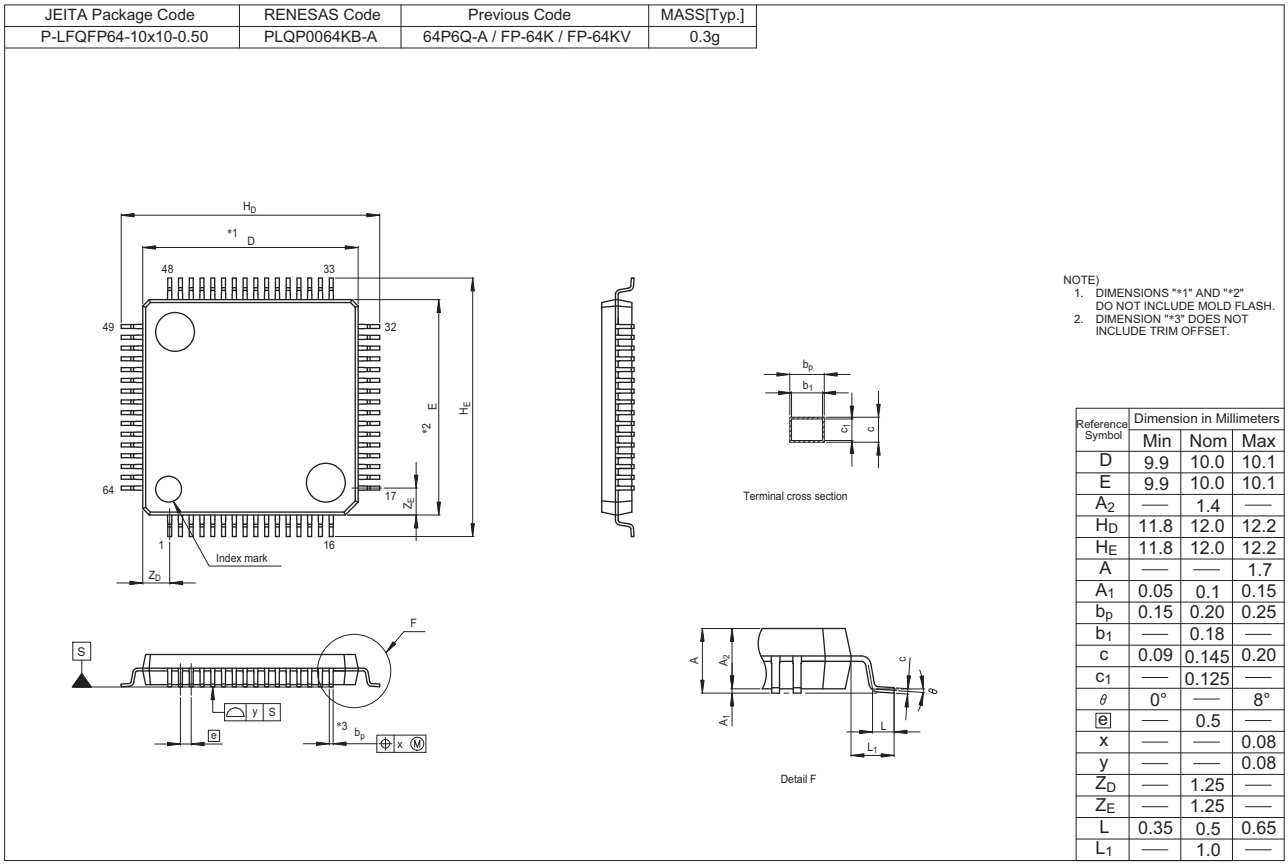
| Port Name (Pin Name) | Reset | Software Standby Mode | |
|---|---|---|---|
| P03 (DA0) | Hi-Z | DA0 output (DAOE0 = 1) | DA output retained |
| | | Other than the above (DAOE0 = 0) | Keep-O |
| P05 (DA1) | Hi-Z | DA1 output (DAOE1 = 1) | DA output retained |
| | | Other than the above (DAOE1 = 0) | Keep-O |
| P14 (OVRCURA/IRQ4) | Hi-Z | Keep-O*1, *2 | |
| P15 (IRQ5/CLKOUT) | Hi-Z | CLKOUT selected | CLKOUT output |
| | | Other than the above | Keep-O*1 |
| P16 (VBUS/OVRCURB/IRQ6/ RTCOUT) | Hi-Z | RTCOUT selected | RTCOUT output |
| | | Other than the above | Keep-O*1, *2, *3 |
| P17 (IRQ7) | Hi-Z | Keep-O*1 | |
| P26 | Hi-Z | Keep-O | |
| P27 (IRQ3) | Hi-Z | Keep-O*1 | |
| P30 (IRQ0) | Hi-Z | Keep-O*1 | |
| P31 (IRQ1) | Hi-Z | Keep-O*1 | |
| P32 (IRQ2/RTCOUT) | Hi-Z | RTCOUT selected | RTCOUT output |
| | | Other than the above | Keep-O*1 |
| P35 (NMI) | Hi-Z | Keep*1 | |
| P40 to P44, P46 | Hi-Z | Keep-O | |
| P54, P55 | Hi-Z | Keep-O | |
| PA0 | Hi-Z | Keep-O | |
| PA1 (RTCOUT) | Hi-Z | RTCOUT selected | RTCOUT output |
| | | Other than the above | Keep-O |
| PA3 (IRQ6) | Hi-Z | Keep-O*1 | |
| PA4 (IRQ5) | Hi-Z | Keep-O*1 | |
| PA6 (IRQ3) | Hi-Z | Keep-O*1 | |
| PB0 (IRQ2) | Hi-Z | RTCOUT selected | RTCOUT output |
| | | Other than the above | Keep-O*1 |
| PB1 (IRQ4) | Hi-Z | Keep-O*1 | |
| PB3 (OVRCURA) | Hi-Z | Keep-O*2 | |
| PB5 to PB7 | Hi-Z | Keep-O | |
| PC2, PC3 | Hi-Z | Keep-O | |
| PC4 (VBUS/IRQ2/CLKOUT) | Hi-Z | CLKOUT selected | CLKOUT output |
| | | Other than the above | Keep-O*1, *2 |
| PC5 | Hi-Z | Keep-O | |
| PC6 | Hi-Z | Keep-O | |
| PC7 (OVRCURB) | Hi-Z | Keep-O*2 | |
| PE0 (IRQ0) | Hi-Z | Keep-O*1 | |
| PE1 (IRQ1) | Hi-Z | Keep-O*1 | |
| PE2 (IRQ7) | Hi-Z | Keep-O*1 | |
| PE3 (IRQ3) | Hi-Z | Keep-O*1 | |
| PE4 (IRQ4) | Hi-Z | Keep-O*1 | |
| PE5 (IRQ5) | Hi-Z | Keep-O*1 | |
| PE6 (IRQ6) | Hi-Z | Keep-O*1 | |
| PE7 (IRQ7) | Hi-Z | Keep-O*1 | |
| PH7 | Hi-Z | Keep | |
| PJ6 | Hi-Z | Keep-O | |
| PJ7 | Hi-Z | Keep-O | |

H: High level
L: Low level
Keep-O: Output pins retain their previous values, and input pins become high-impedance.
Keep: Pin states are retained during periods in software standby mode. (pull-up and open-drain settings are also retained.)
Hi-Z: High-impedance
Note 1. Input is enabled if the pin is specified as the software standby mode canceling source while it is used as an external interrupt pin.
Note 2. Input is enabled while the pin is used as the USB pin (VBUS/OVRCURA/OVRCURB).
Note 3. Do not input or output a high-level signal in software standby mode if the pin is selected as a peripheral function or external interrupt pin.

# Appendix 2. Package Dimensions

Information on the latest version of the package dimensions or mountings has been displayed in “Packages” on Renesas Electronics Corporation website.

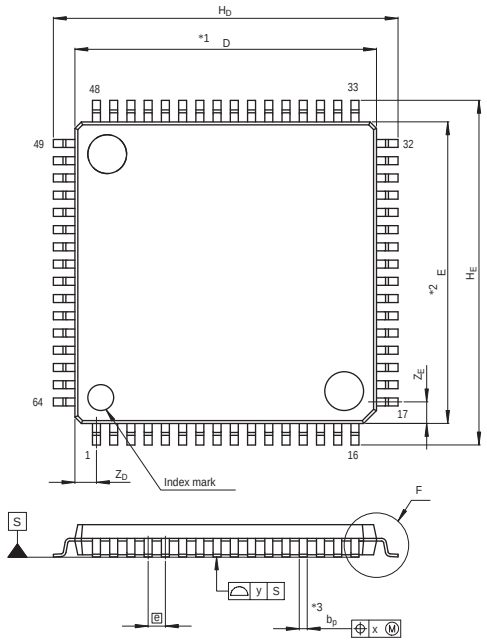
| JEITA Package Code | RENESAS Code | Previous Code | MASS[Typ.] |
|---|---|---|---|
| P-LFQFP64-10x10-0.50 | PLQP0064KB-A | 64P6Q-A / FP-64K / FP-64KV | 0.3g |

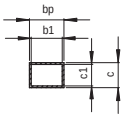



| Reference Symbol | Dimension in Millimeters | | |
|---|---|---|---|
| | Min | Nom | Max |
| D | 9.9 | 10.0 | 10.1 |
| E | 9.9 | 10.0 | 10.1 |
| $A_2$ | — | 1.4 | — |
| $H_D$ | 11.8 | 12.0 | 12.2 |
| $H_E$ | 11.8 | 12.0 | 12.2 |
| A | — | — | 1.7 |
| $A_1$ | 0.05 | 0.1 | 0.15 |
| $b_p$ | 0.15 | 0.20 | 0.25 |
| $b_1$ | — | 0.18 | — |
| c | 0.09 | 0.145 | 0.20 |
| $c_1$ | — | 0.125 | — |
| θ | 0° | — | 8° |
| e | — | 0.5 | — |
| x | — | — | 0.08 |
| y | — | — | 0.08 |
| $Z_D$ | — | 1.25 | — |
| $Z_E$ | — | 1.25 | — |
| L | 0.35 | 0.5 | 0.65 |
| $L_1$ | — | 1.0 | — |

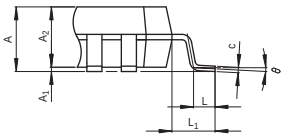
Figure A 64-Pin LQFP (PLQP0064KB-A)

| JEITA Package Code | RENESAS Code | Previous Code | MASS[Typ.] |
|---|---|---|---|
| P-LQFP64-14x14-0.80 | PLQP0064GA-A | 64P6U-A/ — | 0.7g |




Terminal cross section




Detail F

NOTE)
1. DIMENSIONS "*1" AND "*2" DO NOT INCLUDE MOLD FLASH.
2. DIMENSION "*3" DOES NOT INCLUDE TRIM OFFSET.

| Reference Symbol | Dimension in Millimeters | | |
|---|---|---|---|
| | Min | Nom | Max |
| D | 13.9 | 14.0 | 14.1 |
| E | 13.9 | 14.0 | 14.1 |
| $A_2$ | — | 1.4 | — |
| $H_D$ | 15.8 | 16.0 | 16.2 |
| $H_E$ | 15.8 | 16.0 | 16.2 |
| A | — | — | 1.7 |
| $A_1$ | 0 | 0.1 | 0.2 |
| $b_p$ | 0.32 | 0.37 | 0.42 |
| $b_1$ | — | 0.35 | — |
| c | 0.09 | 0.145 | 0.20 |
| $c_1$ | — | 0.125 | — |
| θ | 0° | — | 8° |
| e | — | 0.8 | — |
| x | — | — | 0.20 |
| y | — | — | 0.10 |
| $Z_D$ | — | 1.0 | — |
| $Z_E$ | — | 1.0 | — |
| L | 0.3 | 0.5 | 0.7 |
| $L_1$ | — | 1.0 | — |

Figure B 64-Pin LQFP (PLQP0064GA-A)

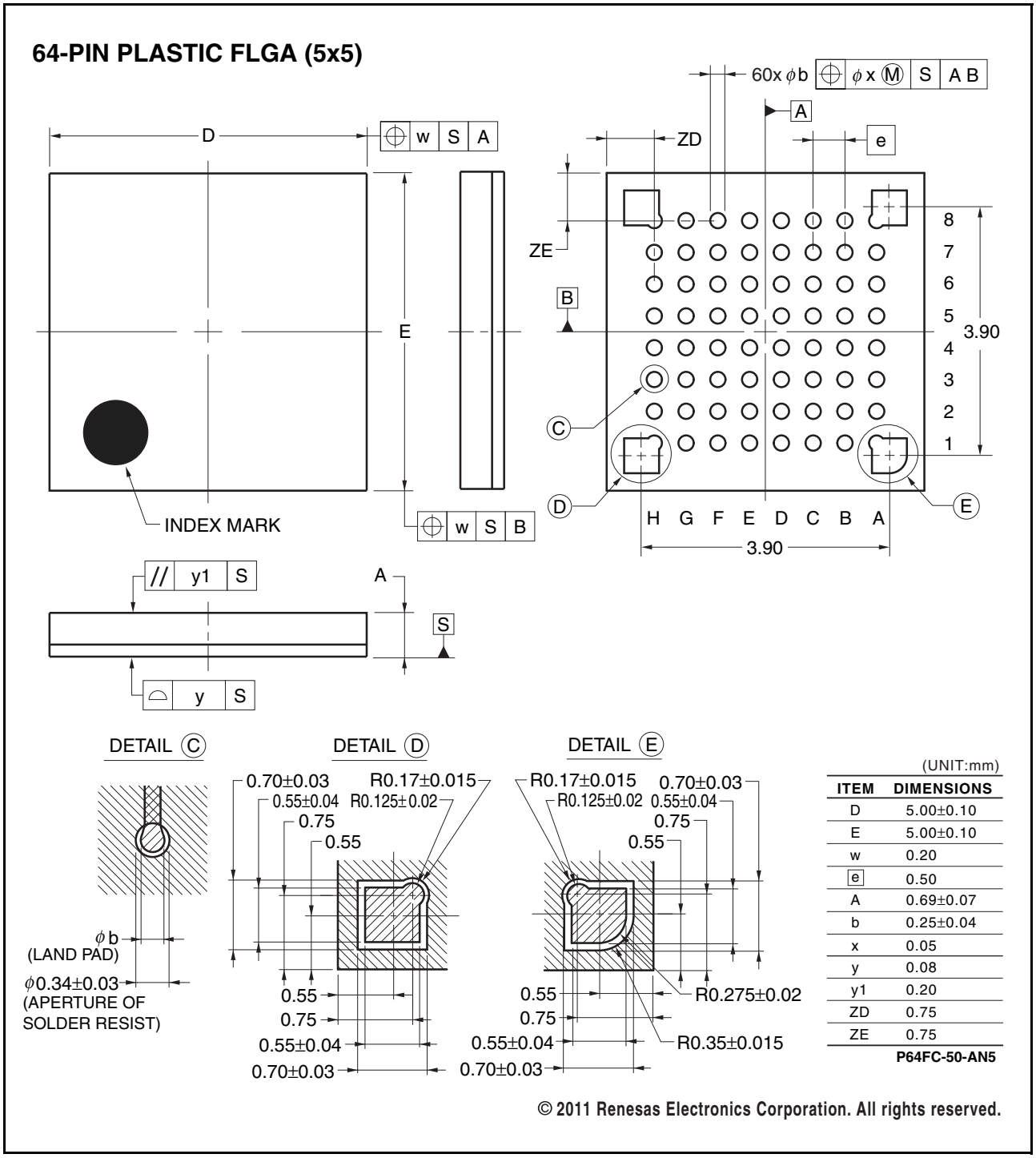
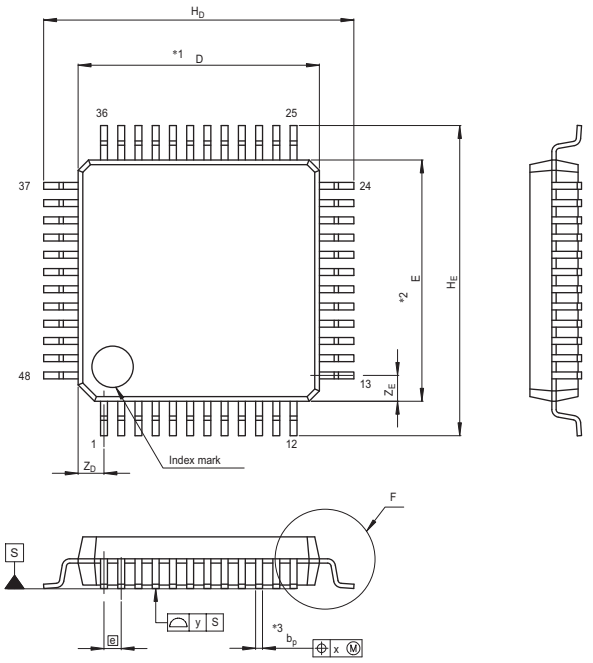
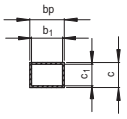



Figure C 64-Pin WFLGA (PWLG0064KA-A)

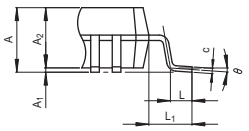
| JEITA Package Code | RENESAS Code | Previous Code | MASS[Typ.] |
|---|---|---|---|
| P-LFQFP48-7x7-0.50 | PLQP0048KB-A | 48P6Q-A | 0.2g |






Terminal cross section



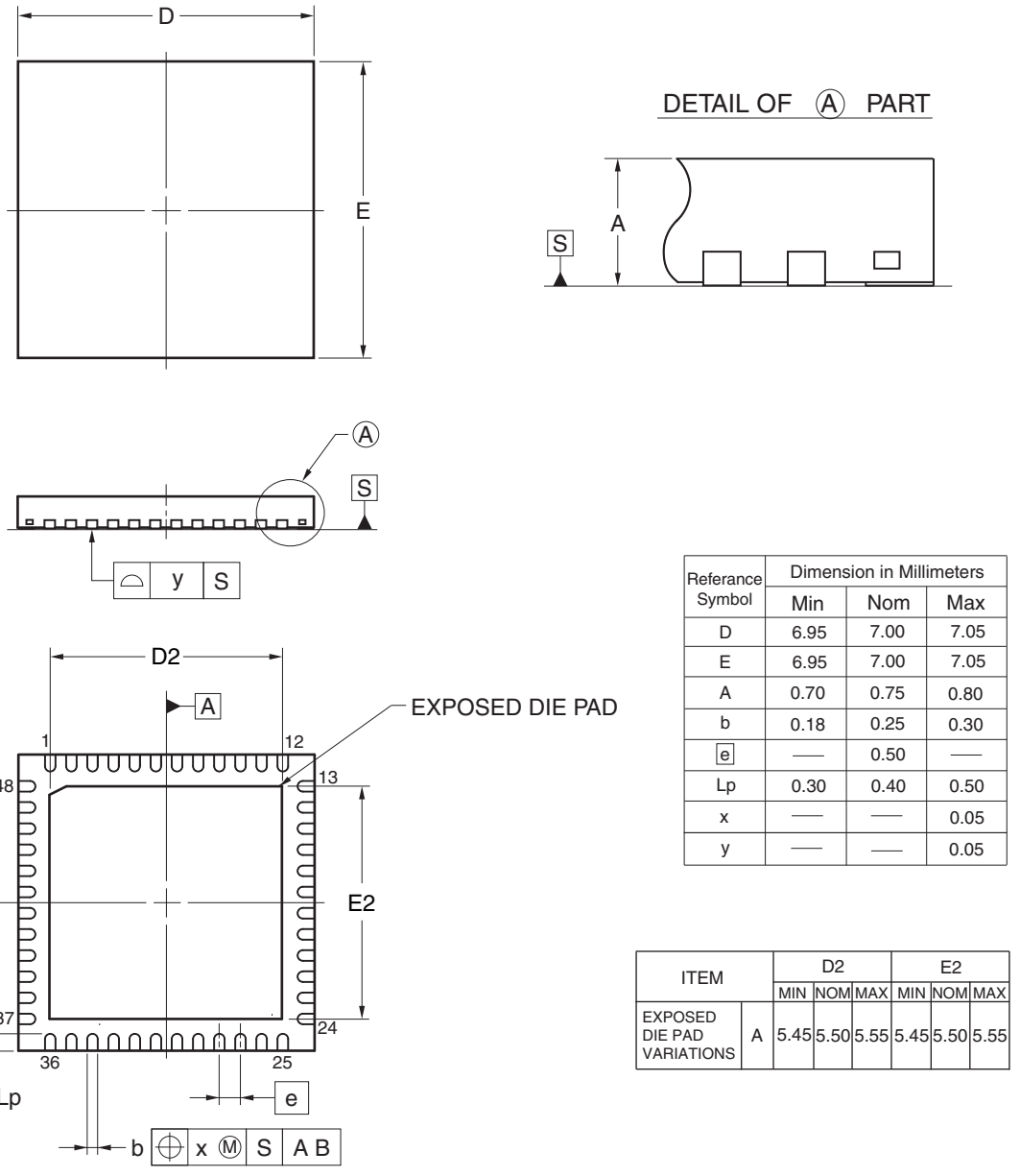

Detail F

NOTE)
1. DIMENSIONS "*1" AND "*2" DO NOT INCLUDE MOLD FLASH.
2. DIMENSION "*3" DOES NOT INCLUDE TRIM OFFSET.

| Reference Symbol | Dimension in Millimeters | | |
|---|---|---|---|
| | Min | Nom | Max |
| D | 6.9 | 7.0 | 7.1 |
| E | 6.9 | 7.0 | 7.1 |
| $A_2$ | — | 1.4 | — |
| $H_D$ | 8.8 | 9.0 | 9.2 |
| $H_E$ | 8.8 | 9.0 | 9.2 |
| A | — | — | 1.7 |
| $A_1$ | 0 | 0.1 | 0.2 |
| $b_p$ | 0.17 | 0.22 | 0.27 |
| $b_1$ | — | 0.20 | — |
| c | 0.09 | 0.145 | 0.20 |
| $c_1$ | — | 0.125 | — |
| θ | 0° | — | 8° |
| e | — | 0.5 | — |
| x | — | — | 0.08 |
| y | — | — | 0.10 |
| $Z_D$ | — | 0.75 | — |
| $Z_E$ | — | 0.75 | — |
| L | 0.35 | 0.5 | 0.65 |
| $L_1$ | — | 1.0 | — |

Figure D 48-Pin LQFP (PLQP0048KB-A)

| JEITA Package Code | RENESAS Code | Previous Code | MASS (TYP.) [g] |
|---|---|---|---|
| P-HWQFN48-7x7-0.50 | PWQN0048KB-A | 48PJN-A<br>P48K8-50-5B4-5 | 0.13 |



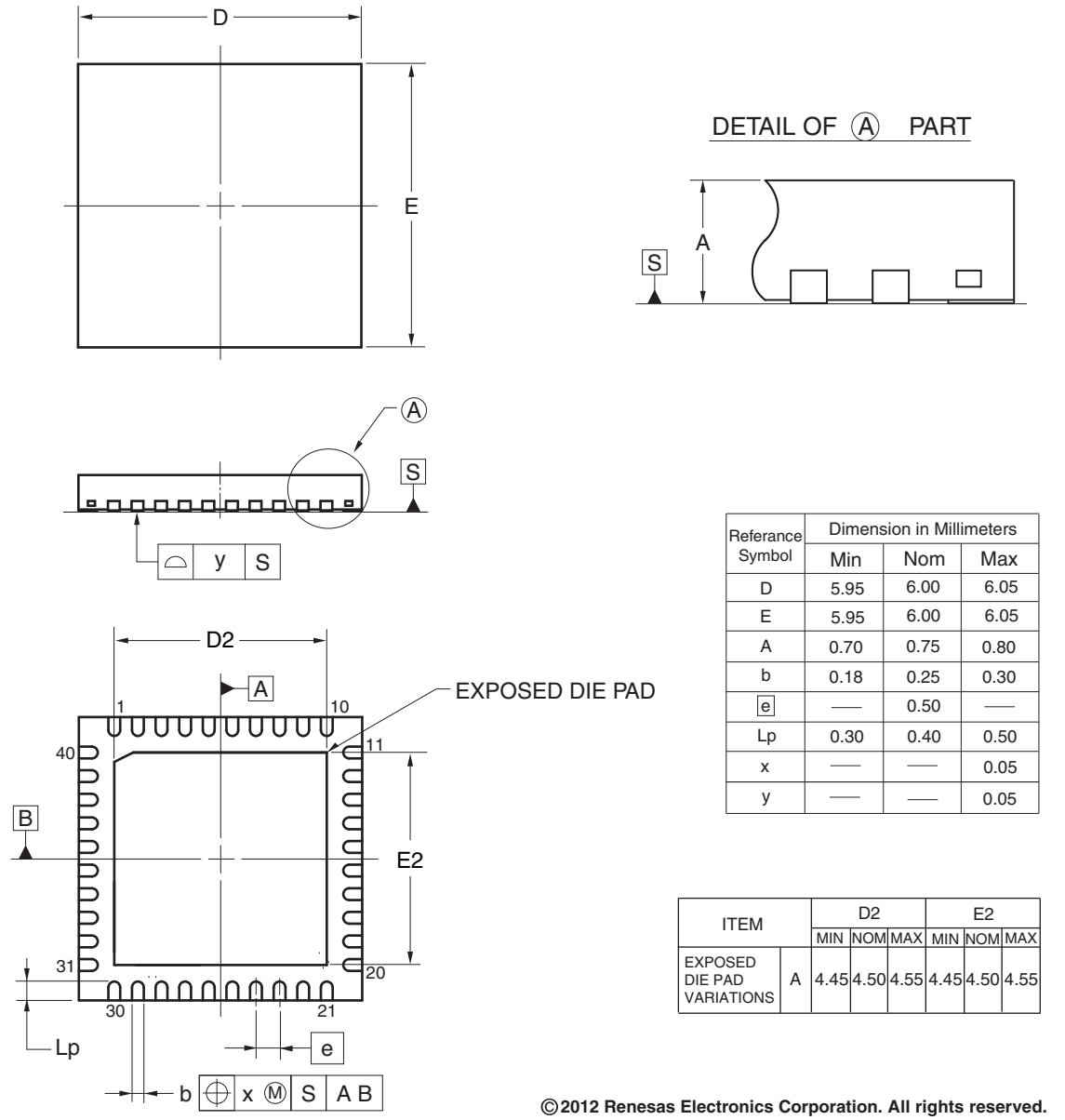

| Referance Symbol | Dimension in Millimeters | | |
|---|---|---|---|
| | Min | Nom | Max |
| D | 6.95 | 7.00 | 7.05 |
| E | 6.95 | 7.00 | 7.05 |
| A | 0.70 | 0.75 | 0.80 |
| b | 0.18 | 0.25 | 0.30 |
| e | — | 0.50 | — |
| Lp | 0.30 | 0.40 | 0.50 |
| x | — | — | 0.05 |
| y | — | — | 0.05 |

| ITEM | | D2 | | | E2 | | |
|---|---|---|---|---|---|---|---|
| | | MIN | NOM | MAX | MIN | NOM | MAX |
| EXPOSED DIE PAD VARIATIONS | A | 5.45 | 5.50 | 5.55 | 5.45 | 5.50 | 5.55 |



Figure E 48-Pin HWQFN (PWQN0048KB-A)

| JEITA Package Code | RENESAS Code | Previous Code | MASS (TYP.) [g] |
|---|---|---|---|
| P-HWQFN40-6x6-0.50 | PWQN0040KC-A | P40K8-50-4B4-4 | 0.09 |



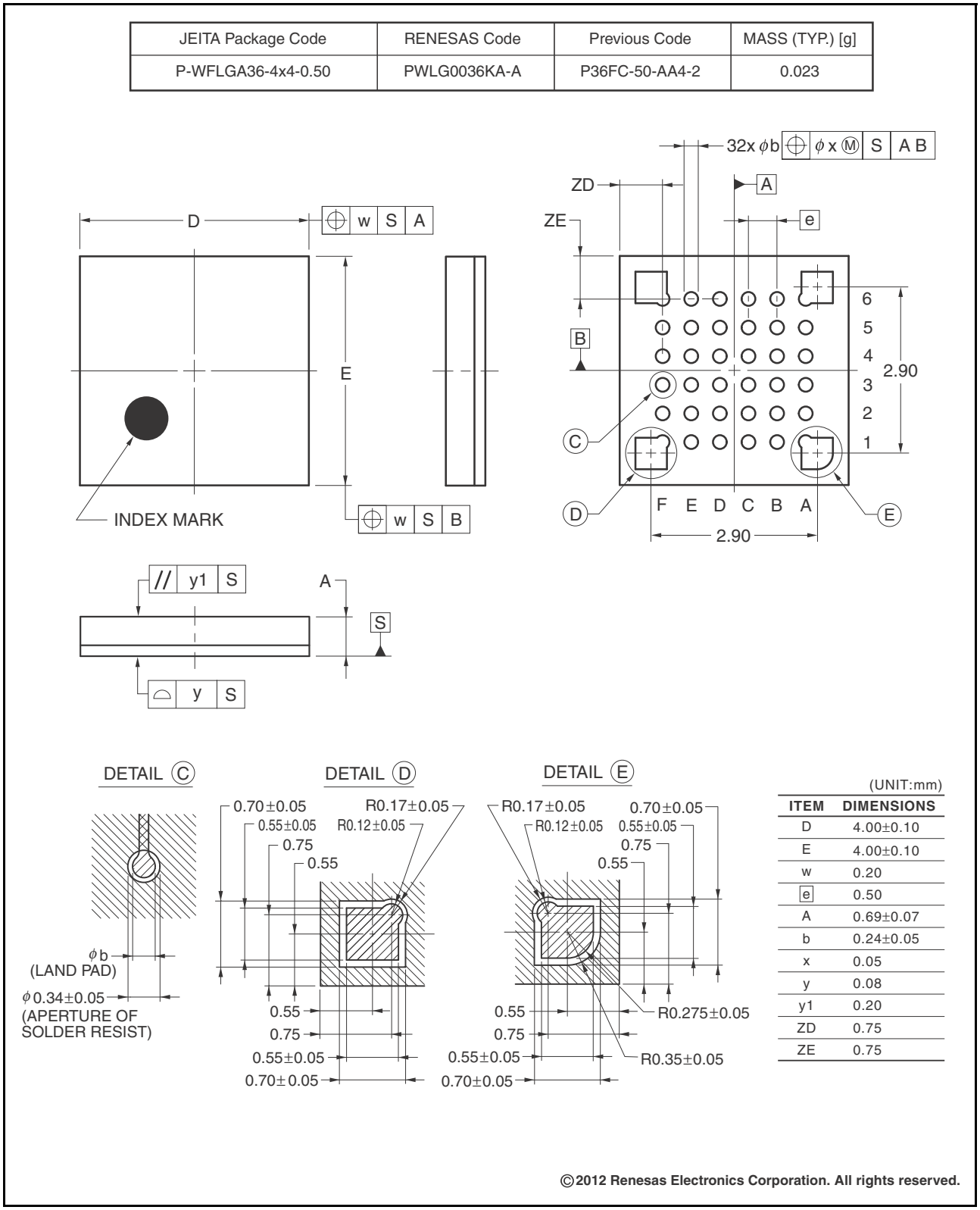

| Referance Symbol | Dimension in Millimeters | | |
|---|---|---|---|
| | Min | Nom | Max |
| D | 5.95 | 6.00 | 6.05 |
| E | 5.95 | 6.00 | 6.05 |
| A | 0.70 | 0.75 | 0.80 |
| b | 0.18 | 0.25 | 0.30 |
| e | — | 0.50 | — |
| Lp | 0.30 | 0.40 | 0.50 |
| x | — | — | 0.05 |
| y | — | — | 0.05 |

| ITEM | | D2 | | | E2 | | |
|---|---|---|---|---|---|---|---|
| | | MIN | NOM | MAX | MIN | NOM | MAX |
| EXPOSED DIE PAD VARIATIONS | A | 4.45 | 4.50 | 4.55 | 4.45 | 4.50 | 4.55 |



Figure F 40-Pin HWQFN (PWQN0040KC-A)

| JEITA Package Code | RENESAS Code | Previous Code | MASS (TYP.) [g] |
|---|---|---|---|
| P-WFLGA36-4x4-0.50 | PWLG0036KA-A | P36FC-50-AA4-2 | 0.023 |




(UNIT:mm)

| ITEM | DIMENSIONS |
|---|---|
| D | 4.00±0.10 |
| E | 4.00±0.10 |
| w | 0.20 |
| e | 0.50 |
| A | 0.69±0.07 |
| b | 0.24±0.05 |
| x | 0.05 |
| y | 0.08 |
| y1 | 0.20 |
| ZD | 0.75 |
| ZE | 0.75 |



Figure G 36-Pin WFLGA (PWLG0036KA-A)

| REVISION HISTORY | RX111 Group User's Manual: Hardware |
|---|---|

| Rev. | Date | Description | |
|---|---|---|---|
| | | Page | Summary |
| 1.00 | Jun 14, 2013 | — | First edition, issued |

RX111 Group User's Manual: Hardware

Publication Date: Rev.1.00 Jun 14, 2013

Published by: Renesas Electronics Corporation

**SALES OFFICES** Renesas Electronics Corporation http://www.renesas.com

Refer to "http://www.renesas.com/" for the latest and detailed information.


**California Eastern Laboratories, Inc.**
4590 Patrick Henry Drive, Santa Clara, California 95054, U.S.A.
Tel: +1-408-919-2500, Fax: +1-408-988-0279

**Renesas Electronics Europe Limited**
Dukes Meadow, Millboard Road, Bourne End, Buckinghamshire, SL8 5FH, U.K
Tel: +44-1628-651-700, Fax: +44-1628-651-804

**Renesas Electronics Europe GmbH**
Arcadiastrasse 10, 40472 Düsseldorf, Germany
Tel: +49-211-65030, Fax: +49-211-6503-1327

**Renesas Electronics (China) Co., Ltd.**
7th Floor, Quantum Plaza, No.27 ZhiChunLu Haidian District, Beijing 100083, P.R.China
Tel: +86-10-8235-1155, Fax: +86-10-8235-7679

**Renesas Electronics (Shanghai) Co., Ltd.**
Unit 204, 205, AZIA Center, No.1233 Lujiazui Ring Rd., Pudong District, Shanghai 200120, China
Tel: +86-21-5877-1818, Fax: +86-21-6887-7858 / -7898

**Renesas Electronics Hong Kong Limited**
Unit 1601-1613, 16/F., Tower 2, Grand Century Place, 193 Prince Edward Road West, Mongkok, Kowloon, Hong Kong
Tel: +852-2886-9318, Fax: +852 2886-9022/9044

**Renesas Electronics Taiwan Co., Ltd.**
13F, No. 363, Fu Shing North Road, Taipei, Taiwan
Tel: +886-2-8175-9600, Fax: +886 2-8175-9670

**Renesas Electronics Singapore Pte. Ltd.**
80 Bendemeer Road, Unit #06-02 Hyflux Innovation Centre Singapore 339949
Tel: +65-6213-0200, Fax: +65-6213-0300

**Renesas Electronics Malaysia Sdn.Bhd.**
Unit 906, Block B, Menara Amcorp, Amcorp Trade Centre, No. 18, Jln Persiaran Barat, 46050 Petaling Jaya, Selangor Darul Ehsan, Malaysia
Tel: +60-3-7955-9390, Fax: +60-3-7955-9510

**Renesas Electronics Korea Co., Ltd.**
11F., Samik Lavied' or Bldg., 720-2 Yeoksam-Dong, Kangnam-Ku, Seoul 135-080, Korea
Tel: +82-2-558-3737, Fax: +82-2-558-5141



Colophon 1.3

# RX111 Group

Renesas Electronics Corporation

R01UH0365EJ0100